CONFIGURATION GUIDE

Extreme NetIron Software Defined Networking (SDN) Guide, 6.3.00a

Supporting NetIron OS 6.3.00a

9035614-01
November 2018

Legal Notice

Extreme Networks, Inc. reserves the right to make changes in specifications and other information contained in this document and its website without prior notice. The reader should in all cases consult representatives of Extreme Networks to determine whether any such changes have been made.

The hardware, firmware, software or any specifications described or referred to in this document are subject to change without notice.

Trademarks

Extreme Networks and the Extreme Networks logo are trademarks or registered trademarks of Extreme Networks, Inc. in the United States and/or other countries.

All other names (including any product names) mentioned in this document are the property of their respective owners and may be trademarks or registered trademarks of their respective companies/owners.

For additional information on Extreme Networks trademarks, please see: www.extremenetworks.com/company/legal/trademarks

Software Licensing

Some software files have been licensed under certain open source or third-party licenses. End-user license agreements and open source declarations can be found at: www.extremenetworks.com/support/policies/software-licensing

Support

For product support, phone the Global Technical Assistance Center (GTAC) at 1-800-998-2408 (toll-free in U.S. and Canada) or +1-408-579-2826. For the support phone number in other countries, visit: http://www.extremenetworks.com/support/contact/

For product documentation online, visit: https://www.extremenetworks.com/documentation/

Contents

Preface

This section discusses the conventions used in this guide, ways to provide feedback, additional help, and other Extreme Networks® publications.

Conventions

This section discusses the conventions used in this guide.

Notes, cautions, and warnings

Notes, cautions, and warning statements may be used in this document. They are listed in the order of increasing severity of potential hazards.

NOTE
A Note provides a tip, guidance, or advice, emphasizes important information, or provides a reference to related information.

ATTENTION
An Attention statement indicates a stronger note, for example, to alert you when traffic might be interrupted or the device might reboot.

CAUTION
A Caution statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data.

DANGER
A Danger statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.

Text formatting conventions

Text formatting conventions such as boldface, italic, or Courier font may be used to highlight specific words or phrases.

Format	Description
bold text	Identifies command names.
	Identifies keywords and operands.
	Identifies the names of GUI elements.
	Identifies text to enter in the GUI.
italic text	Identifies emphasis.
	Identifies variables.
	Identifies document titles.

Format	Description
`Courier font`	Identifies CLI output. Identifies command syntax examples.

Command syntax conventions

Bold and italic text identify command syntax components. Delimiters and operators define groupings of parameters and their logical relationships.

Convention	Description
bold text	Identifies command names, keywords, and command options.
italic text	Identifies a variable.
[ ]	Syntax components displayed within square brackets are optional. Default responses to system prompts are enclosed in square brackets.
{ **x** | **y** | **z** }	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options.
x | **y**	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, for example, passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, *member*[*member*...].
\	Indicates a "soft" line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Documentation and Training

To find Extreme Networks product guides, visit our documentation pages at:

Current Product Documentation	www.extremenetworks.com/documentation/
Archived Documentation (for earlier versions and legacy products)	www.extremenetworks.com/support/documentation-archives/
Release Notes	www.extremenetworks.com/support/release-notes
Hardware/Software Compatibility Matrices	https://www.extremenetworks.com/support/compatibility-matrices/
White papers, data sheets, case studies, and other product resources	https://www.extremenetworks.com/resources/

Open Source Declarations

Some software files have been licensed under certain open source licenses. More information is available at: www.extremenetworks.com/support/policies/open-source-declaration/.

Training

Extreme Networks offers product training courses, both online and in person, as well as specialized certifications. For more information, visit www.extremenetworks.com/education/.

Getting Help

If you require assistance, contact Extreme Networks using one of the following methods:

Extreme Portal	Search the GTAC (Global Technical Assistance Center) knowledge base, manage support cases and service contracts, download software, and obtain product licensing, training, and certifications.
The Hub	A forum for Extreme Networks customers to connect with one another, answer questions, and share ideas and feedback. This community is monitored by Extreme Networks employees, but is not intended to replace specific guidance from GTAC.
Call GTAC	For immediate support: 1-800-998-2408 (toll-free in U.S. and Canada) or +1 408-579-2826. For the support phone number in your country, visit: www.extremenetworks.com/support/contact

Before contacting Extreme Networks for technical support, have the following information ready:

- Your Extreme Networks service contract number and/or serial numbers for all involved Extreme Networks products
- A description of the failure
- A description of any action(s) already taken to resolve the problem
- A description of your network environment (such as layout, cable type, other relevant environmental information)
- Network load at the time of trouble (if known)
- The device history (for example, if you have returned the device before, or if this is a recurring problem)
- Any related RMA (Return Material Authorization) numbers

Subscribing to Service Notifications

You can subscribe to email notifications for product and software release announcements, Vulnerability Notices, and Service Notifications.

1. Go to www.extremenetworks.com/support/service-notification-form.
2. Complete the form with your information (all fields are required).
3. Select the products for which you would like to receive notifications.

   NOTE
   You can modify your product selections or unsubscribe at any time.

4. Click **Submit**.

Providing Feedback to Us

Quality is our first concern at Extreme Networks, and we have made every effort to ensure the accuracy and completeness of this document. We are always striving to improve our documentation and help you work better, so we want to hear from you! We welcome all feedback but especially want to know about:

- Content errors or confusing or conflicting information.
- Ideas for improvements to our documentation so you can find the information you need faster.
- Broken links or usability issues.

If you would like to provide feedback to the Extreme Networks Information Development team, you can do so in two ways:

- Use our short online feedback form at https://www.extremenetworks.com/documentation-feedback/.

- Email us at documentation@extremenetworks.com.

Please provide the publication title, part number, and as much detail as possible, including the topic heading and page number if applicable, as well as your suggestions for improvement.

About This Document

Audience

This document is designed for system administrators with a working knowledge of Layer 2 and Layer 3 switching and routing.

If you are using a Extreme device, you should be familiar with the following protocols if applicable to your network - IP, RIP, OSPF, BGP, ISIS, IGMP, PIM, MPLS, and VRRP.

Supported hardware and software

End of Support for ExtremeSwitching CES 2000 Series devices

Beginning with NetIron OS 6.3.00a and later, the ExtremeSwitching CES 2000 Series devices are not supported. Refer to the End of Sale and End of Support page for additional information.

The hardware platforms in the following table are supported by this release of this guide.

TABLE 1 Supported devices

ExtremeRouting XMR Series	ExtremeRouting MLX Series	ExtremeRouting CER 2000 Series
XMR 4000	MLX-4	CER 2024C
XMR 8000	MLX-8	CER-RT 2024C
XMR 16000	MLX-16	CER 2024F
XMR 32000	MLX-32	CER-RT 2024F
	MLXe-4	CER 2048C
	MLXe-8	CER-RT 2048C
	MLXe-16	CER 2048CX
	MLXe-32	CER-RT 2048CX
		CER 2048F
		CER-RT 2048F
		CER 2048FX
		CER-RT 2048FX

How command information is presented in this guide

Starting with Extreme NetIron 5.6.00, command syntax and parameter descriptions are removed from commands that are referenced in configuration tasks. To find the full description of a specific command, including all required and optional keywords and variables, refer to the *Extreme NetIron Command Reference* for your software release.

What's new in this document

NOTE

The NetIron 6.3.00 release (the image files and the documentation) is no longer available from the Extreme Portal. New software features introduced in release 6.3.00 are included in release 6.3.00a.

On October 30, 2017, Extreme Networks, Inc. acquired the data center networking business from Brocade Communications Systems, Inc. This document has been updated to remove or replace references to Brocade Communications, Inc. with Extreme Networks, Inc., as appropriate.

For the complete list of supported features and the summary of enhancements and configuration notes for this release, refer to the Extreme NetIron OS Release Notes.

OpenFlow v1.0.0

Overview of OpenFlow v1.0.0

An OpenFlow-enabled router supports an OpenFlow Client (control plane software), which communicates with an OpenFlow Controller using OpenFlow. The OpenFlow Controller runs on a server or a server cluster. OpenFlow-enabled routers support the abstraction of a flow table, which is manipulated by the OpenFlow Controller. The flow table contains flow entries. Each flow entry represents a flow (that is, packets with a given MAC address, VLAN tag, IP address, or TCP/UDP port, and so on). The flow table is sorted by flow priority, which is defined by the OpenFlow Controller. The highest priority flows are at the top of the flow table.

Incoming packets on an OpenFlow-enabled port are matched (in order of priority) against the flow entries defined for that port by the OpenFlow Controller. If the packet matches a given flow entry, the flow-matching process stops, and the set of actions defined for that flow entry are performed. Packets that do not match any flow entry are dropped by default. The Extreme implementation of OpenFlow supports an option to send such packets to the OpenFlow Controller. Refer to Configuring the default action on page 32.

FIGURE 1 OpenFlow-enabled router

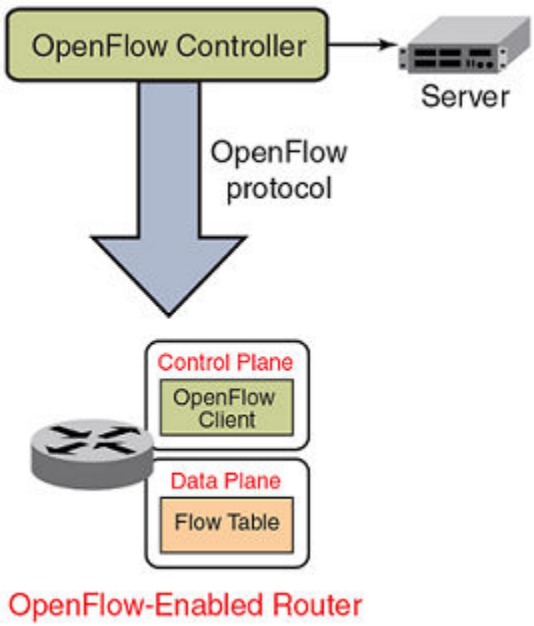
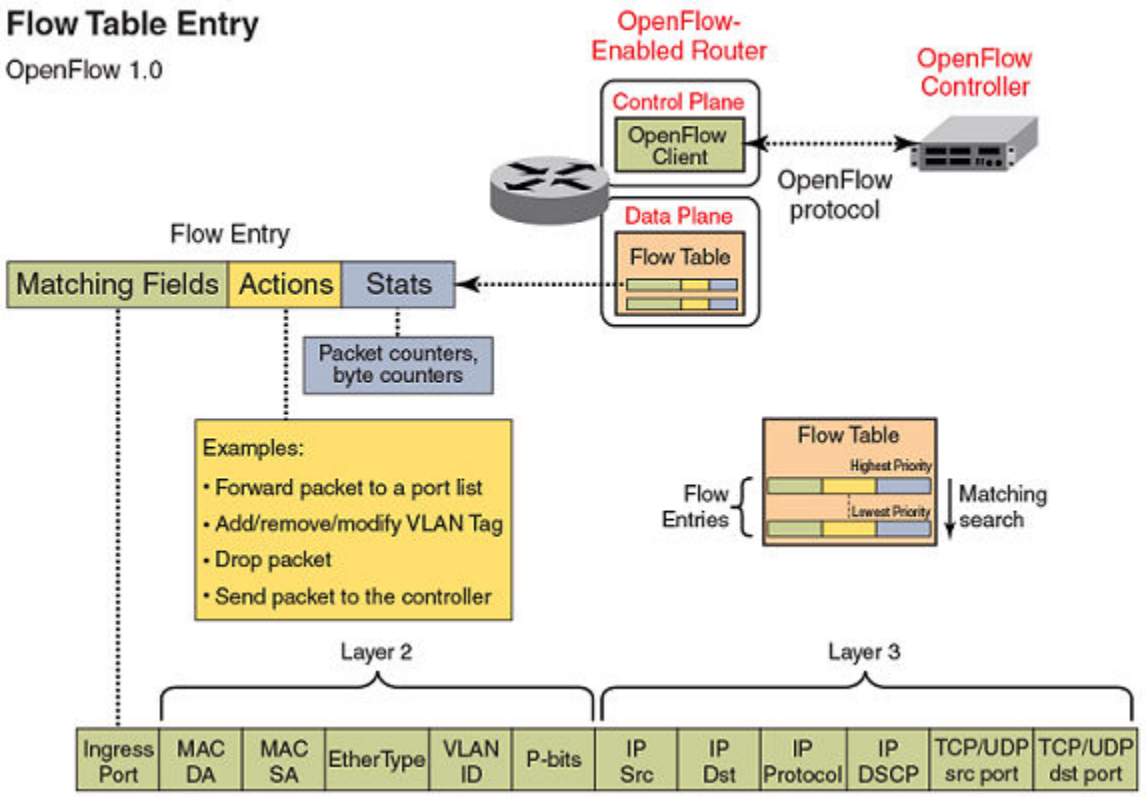


FIGURE 2 OpenFlow flow table entries



Flow table entries

The OpenFlow match rules in the following table are supported on devices for Flow table entries.

The implementation of OpenFlow supports three modes of operation when enabling OpenFlow on a port: Layer 2 mode, Layer 3 mode and Layer23 mode. Layer 2 mode supports OpenFlow matching rules based on the Layer 2 fields shown in Overview of OpenFlow v1.0.0 on page 13, while Layer 3 mode supports the OpenFlow matching rules based on the Layer 3 fields. Layer23 mode supports the OpenFlow matching rules based on the Layer 2 and Layer 3 fields.

The MLX Series and XMR Series devices support enabling ports in either Layer 2 or Layer 3 mode. The CER 2000 Series and CES 2000 Series devices support Layer 2 mode by default (OpenFlow Layer 3 mode configuration on a port is currently not supported on these devices).

TABLE 2 OpenFlow match rules

Match rule	MLX Series XMR Series	CER 2000 Series CES 2000 Series
Port enabled for Layer 2 mode	Yes	Yes
Source port	Yes	Yes
Source or destination MAC address	Yes These devices support either source or destination MAC address, or a combination of both source and destination MAC addresses as the match rule.	These devices support either source MAC or destination MAC address as the match rule, not both at the same time. This is a global option to be configured by the user.

TABLE 2 OpenFlow match rules (continued)

Match rule	MLX Series XMR Series	CER 2000 Series CES 2000 Series
Ether type	Yes	Yes
VLAN ID	Yes	Yes
VLAN priority	Yes	Yes
Untagged packets	Yes	Yes
Port enabled for Layer 3 mode	Yes	No
Ether type (Supports matching to Ether type value 0x88CC LLDP only; supports only 'send to controller' as the action)	Yes	No
Ether type (Supports matching to Ether type ARP value; supports only 'send to controller' as the action)	Yes	No
Source port	Yes	No
VLAN ID	Yes	No
VLAN priority	Yes	No
Source IP address	Yes	No
Destination IP address	Yes	No
Protocol type	Yes	No
IP TOS bits	Yes	No
TCP or UDP source port	Yes	No
Port enabled for Layer23 mode	Yes	Yes
Ether type (Supports matching to Ether type value 0x88CC LLDP only; supports only 'send to controller' as the action)	Yes	Yes
Ether type (Supports matching to Ether type ARP value; supports only 'send to controller' as the action)	Yes	Yes
Source port	Yes	Yes
VLAN ID	Yes	Yes
VLAN priority	Yes	Yes
Source IP address	Yes	Yes
Destination IP address	Yes	Yes
Protocol type	Yes	Yes
IP TOS bits	Yes	Yes
TCP or UDP source port	Yes	Yes
TCP or UDP destination port	Yes	Yes

OpenFlow actions

Each OpenFlow flow table entry contains the list of actions to be performed when a packet matches the flow entry. These actions are defined by the OpenFlow Controller.

Packets that do not match any flow entry are dropped by default. The Extreme implementation of OpenFlow supports an option to send such packets to the OpenFlow Controller. Refer to Configuring the default action on page 32.

Devices support the actions listed in the following table.

TABLE 3 OpenFlow actions supported on Extreme devices

OpenFlow action	MLX Series XMR Series	CER 2000 Series CES 2000 Series
Forward a packet to a port or set of ports	Yes	Yes
Drop the packet	Yes	Yes
Add, modify, or remove VLAN ID or priority on a per destination port basis	Yes	Yes
Modify the IP DSCP (For a flow sending a copy of the packet to multiple destinations, the DSCP modification must be the same for all destinations. Modifying IP DSCP is only supported on ports enabled with Layer 3 mode.)	Yes	No
Modify the source or destination MAC address (for a flow sending a copy of the packet to multiple destinations, the MAC address modification must be the same for all destinations).	Yes	No
Send the packet to the controller (Packet In)	Yes	Yes
Receive the packet from the controller and send it to ports (Packet Out)	Yes	Yes

OpenFlow Controller

Multiple controller connections can be used for redundancy purposes, such as when using a single controller with multiple addresses. Multiple controller connections can also be used to support active-standby controllers.

Regardless of the intended use of multiple controller connections, the device allows all the controller connections to concurrently manage the flow table. That is, flow entries in the flow table are not identified as belonging to any specific controller connection. In an active-standby controller deployment, controllers themselves must coordinate their actions and active-standby states. The device responds to all connected controllers without distinction.

The device supports two types of controller connections (also called modes): active and passive. An active connection is one for which the device initiates (seeks) the TCP connection to a given OpenFlow Controller address. With a passive connection, the device passively waits for the controller to initiate (seek) the TCP connection to the device. Active mode is commonly used with production controllers, while passive mode is commonly used for testing purposes in experimental environments. Optionally, a controller connection can also use SSL encryption.

OpenFlow counters

MLX Series and XMR Series devices record the number of received packets and bytes on a per-flow basis. However, recording the number of received bytes on a per-flow basis is only supported on ports on the 8x10G or 100G cards. CER 2000 Series and CES 2000 Series devices record only the received packets on a per-flow basis--these devices do not record the number of bytes per flow.

The following per-port counters are available in the flow table:

TABLE 4 OpenFlow counters supported on devices

Counter	Description	MLX Series XMR Series	CER 2000 Series CES 2000 Series
Received packets	Number of packets received on the port	Yes	Yes
Transmitted packets	Number of packets transmitted from the port	Yes	Yes
Received bytes	Number of bytes received on the port	Yes	No
Transmitted bytes	Number of bytes transmitted from the port	Yes	Yes
Receive drops	Number of received packets dropped on the port because the packets did not match any rules	Yes	Yes
Transmit drops	Number of transmit packets dropped on the egress port	Yes	Yes
Receive errors	Number of errors detected on the port on received packets	Yes	Yes
Transmit errors	Number of errors detected on the port on transmitted packets	Yes	Yes
Receive frame alignment errors	Number of frame alignment errors detected on packets received on the port	Yes	No
Receive overrun errors	Number of packets that caused overrun in the receive buffer on the port	Yes	No
Receive CRC errors	Number of packets received on the port that had CRC errors	Yes	No
Collisions	Number of collisions recorded on the port	Yes	No

The following table lists the per-flow counters available:

TABLE 5 Per-flow OpenFlow counters supported

Counter	Description	MLX Series XMR Series	CER 2000 Series CES 2000 Series
Received packets	Number of packets received per flow	Yes	Yes
Received bytes	Number of bytes received per flow	Only on 8x10G or 100G cards	No

Considerations and limitations for configuring OpenFlow

Consider the following points when you configure OpenFlow on devices:

- OpenFlow must be enabled globally on the device before you can enable interfaces for OpenFlow.

- You must explicitly enable or disable OpenFlow on each interface using the CLI commands. You cannot use a range of ports to enable OpenFlow on the interface.
- Before you can disable OpenFlow globally on the device, you must disable OpenFlow on all interfaces individually.
- OpenFlow port configuration is not supported on interface carrying GRE traffic.
- Spanning Tree Protocol (STP) and other Layer 2 or Layer 3 protocols are not supported on OpenFlow-enabled ports.
- OpenFlow supports up to three concurrent sessions with a maximum of two concurrent SSL sessions.
- OpenFlow supports up to 3000 configured flows are supported if all the flows are with a wildcard for the incoming port.
- Layer 2 unicast and multicast packets are flooded in the VLAN for protected VLANs and for unprotected VLANs in absence of flows on hybrid OpenFlow ports.
- Local and normal actions defined by the OpenFlow v1.0.0 protocol are not supported.
- After a reboot, a delay of 16 seconds for CER 2000 Series and CES 2000 Series devices, and 180 seconds for MLX Series and XMR Series devices is added before any flows can be pushed. This is the default delay time to allow all modules and ports to come up.
- 8x10GbE and 2x100GbE interface modules support all generic Ether type value in Layer 2 mode. All other interface modules support Ether type 0x88CC (LLDP) only in Layer 2 mode.
- OpenFlow is an ingress feature. The local device generates protocol messages (such as PIM and OSPF) on OpenFlow enabled-ports, if configured, but control packets OpenFlow default rule. Because of this limitation, the PIM neighbor (if configured) comes up on the peer, and multicast traffic hits the OpenFlow interface in all PIM DMs. In a PIM SM, the OpenFlow port connects to an IGMP snooping-enabled LAN that has the multicast source connected.
- On OpenFlow-enabled ports, packets that do not match any flow entry are dropped by default. The Extreme implementation supports an option to send such packets to the OpenFlow Controller. Refer to Configuring the default action on page 32.

  NOTE
  OpenFlow configuration cannot co-exist with IP multicast routing as both use session CAM. IP multicast packets on OpenFlow enabled interfaces is punted to the CPU.

CER 2000 Series and CES 2000 Series devices

- Enabling ports for Layer 3 flows is not supported.
- Either Source MAC address or Destination MAC address can be used as a matching rule. The choice is a device-level configuration that you specify at the time of enabling OpenFlow on the device.
- Statistics are available only for up to 2048 flows on a first-come-first-served basis.

MLX Series and XMR Series devices

- Even though modification of the source and destination MAC address fields of a packet is supported, modification on a per-destination port basis for multi-destination flows is not supported.

  In other words, sending a packet to multiple ports and having the MAC addresses modified on a per-destination port basis is not supported. However, sending a packet to multiple ports and having the MAC addresses modified in the same way for all destination ports is supported. Similar restriction applies to modifications of the DSCP field of packets.
- Modification of VLAN tag (such as adding, removing, modifying, or modifying the PCP bits of a tag) is supported on a per-destination port basis for tagged packets. A flow may specify sending a copy of a packet to multiple destination ports and have each copy of the packet tagged with a different VLAN ID.

At the same time, if the rule is to send a tagged packet to multiple outgoing ports with different VLAN tags, and if the action for one of the outgoing ports is to send the packet as untagged, the packet is still tagged with the VLAN ID of 0 on that port. For egress ports with the action specifying modified VLAN tag, the specified VLAN tag itself is added.

- An action to modify the VLAN priority of an untagged frame will result in a VLAN tag being added to the frame. If the action does not specify a VLAN ID value, the VLAN ID will be set to 0.
- The action to modify IP DSCP is only supported for flows on ports enabled for Layer 3 mode.
- Matching a VLAN ID on a Layer 3 mode port is only supported for packets with an IP payload.

OpenFlow hybrid switch mode and OpenFlow hybrid port mode

Hybrid switch mode

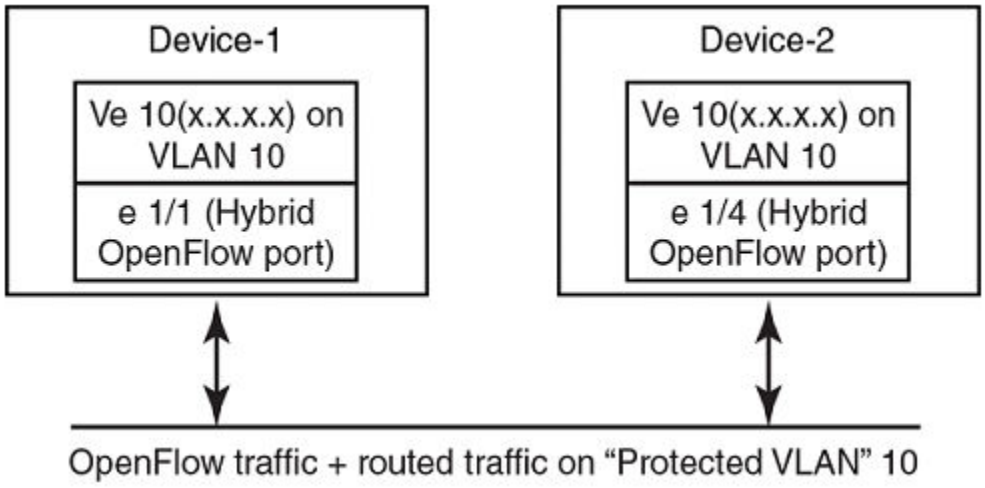
The device supports enabling OpenFlow on a per-port basis, so you can choose which ports of the device will be controlled by the OpenFlow feature. Non-OpenFlow-enabled ports continue to support existing features of the device, such as MPLS, VPLS, IPv4 or IPv6 routing for Layer 2 switching.

OpenFlow hybrid port mode

OpenFlow hybrid-enabled ports support both OpenFlow traffic forwarding and normal routing traffic forwarding. OpenFlow hybrid-enabled ports support "protected VLANs" and "unprotected VLANs". Protected VLANs are not subject to defined OpenFlow flows on the OpenFlow hybrid-enabled ports. OpenFlow flows on a hybrid-enabled port will not match any traffic on protected VLANs. Unprotected VLANs are subject to defined OpenFlow flows on the OpenFlow hybrid-enabled port. OpenFlow flows on a hybrid-enabled port are allowed to match on the traffic of unprotected VLANs.

Figure 3 shows a topology in which port 1/1 on Device-1 and port 1/4 on Device-2 are hybrid-enabled OpenFlow ports with VLAN 10 as a configured protected VLAN. By configuring a virtual Ethernet (VE) interface on a protected VLAN 10 and assigning an address to route the traffic of the nodes, you are able to send protected VLAN traffic between the nodes and route the traffic as per the VE interface. Traffic flowing on other VEs created on top of other VLANs (the unprotected VLANs) is treated as unprotected VLAN traffic and is subject to OpenFlow rules lookup. OpenFlow traffic can be forwarded through this port.

FIGURE 3 OpenFlow hybrid port mode topology



OpenFlow hybrid port mode operation

Consider Device-1 in OpenFlow hybrid port mode on page 19. Ingress traffic on VLAN 10 on hybrid port 1/1 is processed for IPv4 and IPv6 unicast routing. Traffic on other VLANs is processed against OpenFlow flows on port 1/1 and switched accordingly. A preconfigured number of protected VLANs can be supported for normal routing. The Spanning Tree Protocol (STP) state of these routing VLANs is set to forwarding, as the Layer 2 protocol is not supported.

Configuring OpenFlow hybrid port mode

1. Enable OpenFlow at the global configuration level.
2. Configure OpenFlow controller configurations.
3. Configure the system maximum configuration for the maximum OpenFlow entries. (The default is 0.)
4. Configure the maximum OpenFlow flow-protected VLAN entries. (The default is 0.)

   NOTE
   System reload is required once you change the system maximum values.

5. Configure the maximum OpenFlow unprotected VLAN entries. (The default is 0.)
6. Configure protected VLANs on the port. A maximum of 40 protected VLANs can be configured on an OpenFlow port.
7. Enable OpenFlow hybrid port mode on the desired interfaces.
8. Configure a VE for the interface by specifying the protected VLAN and add routing entries.

Feature information

- Switchover and HLOS are not supported. When the active management processor (MP) goes down, communication with the controller is brought down and the flow tables on the MP and all line processors (LP) are cleared. The connection with the controller is re-established after switchover.
- When LP is reset, the flow table on the LP is restored once the LP comes up and flows specific to that LP are maintained in the MP.

- When an OpenFlow-enabled port goes up or down, no rules are removed. The addition or deletion of rules depends solely on the controller.
- 4K OpenFlow entries are supported.
- Up to 2K protected VLANs per system are supported.

Capabilities and prerequisites

The following are current capabilities and prerequisites of OpenFlow hybrid port mode:

- IPv4 and IPv6 unicast routing are supported on OpenFlow protected and unprotected VLANs.
- Packets tagged with a protected VLAN ID are forwarded by IPv4 and IPv6 unicast routing, if IPv4 or IPv6 routing is configured on that VLAN. If IPv4 or IPv6 routing is not configured on that VLAN, such packets drop.
- Packets tagged with an unprotected VLAN ID are subject first to OpenFlow flows. If there is a match on an OpenFlow flow, the packet is forwarded according to the flow actions. No further IPv4 or IPv6 routing is supported for packets that are forwarded by OpenFlow flows. If there is no match on any OpenFlow flow, the packet is forwarded by IPv4 or IPv6 unicast routing, if IPv4 or IPv6 routing is configured on the VLAN. If IPv4 or IPv6 routing is not configured on the VLAN, those packets are either dropped or sent to the controller, per the OpenFlow configuration.
- Layer 2 or L2VPN forwarding is not supported on ports in hybrid-enabled ports because MAC learning is disabled on these ports.
- A port can be enabled for OpenFlow hybrid port mode only if the port is untagged in the default VLAN.
- Ports in OpenFlow hybrid port mode cannot be added as untagged ports to regular VLANs or L2VPN because this can cause a problem with topology discovery.
- Untagged traffic on protected VLAN or unprotected VLAN are supported on OpenFlow hybrid port.
- As routing is enabled on a port in OpenFlow hybrid port mode, OpenFlow traffic or unprotected VLAN traffic sent with the destination MAC address as the port's MAC address and matching IP route entries on the port can potentially find the VLAN and MAC address modified unless the OpenFlow rules explicitly set the VLAN and destination MAC address in the outgoing packet.
- Inbound normal ACL configuration is not supported on the port in hybrid port mode.
- Any port with the default VLAN not equal to the system default VLAN ID cannot be enabled for hybrid port mode.
- Policy-based routing (PBR) is not supported.
- Protected VLAN traffic that does not have matching IP route entries is dropped.
- Multiple interfaces cannot be part of a VE interface created on a port in OpenFlow hybrid port mode with a protected VLAN.
- The BGP, OSPF, and IS-IS protocols are supported on protected VLANs:

  NOTE
  Layer 2 or L2VPN, VRF are not supported.

- When protected VLANs are configured but the port is not part of the VLAN, the traffic coming on the port with the protected VLAN is dropped.
- Port in hybrid-enabled OpenFlow doesn't support MPLS running on the same port.
- Link aggregation is not supported.

Enabling OpenFlow hybrid port mode

Use the **openflow enable** command to enable or disable OpenFlow hybrid port mode on the port and the port becomes a normal port on an interface. The **no** form of the command disables the OpenFlow hybrid port mode on the port and the port becomes a normal port.

```
device(config-if-e10000-2/5)# openflow enable layer2 hybrid-mode
```

To configure IPv6 port, use the following command for enabling OpenFlow hybrid port mode.

```
device # show run interface e1/4

interface ethernet 1/4
device(config-if-e10000-1/4)# openflow enable layer23 hybrid-mode ipv6-match
```

Adding or deleting protected VLANs

Use the **openflow protected -vlans** command to add or delete protected VLANs on an OpenFlow hybrid port mode interface. The **no** form of the command deletes the configured protected VLANs from the hybrid-enabled port.

```
device(config-if-e10000-2/5)# openflow protected-vlans 10
```

VLANs can be configured individually.

NOTE
You cannot specify a VLAN range for the **openflow protected-vlans** command.

Rate limiting capabilities on OpenFlow enabled ports

Rate-limiting support on OpenFlow enabled ports:

- OpenFlow ports (non-hybrid port mode ports): Normal CLI configuration of port-based rate limiting is supported. Per VLAN rate limiting not supported.
- You must set the rate control for certain protocols at the global configuration level.

NOTE
Rate control for certain protocols, such as ARP, is based on the global configuration. For a very large burst of ARP traffic on a port, the system may become unresponsive.

VPLS support for VLANs on OpenFlow hybrid mode ports

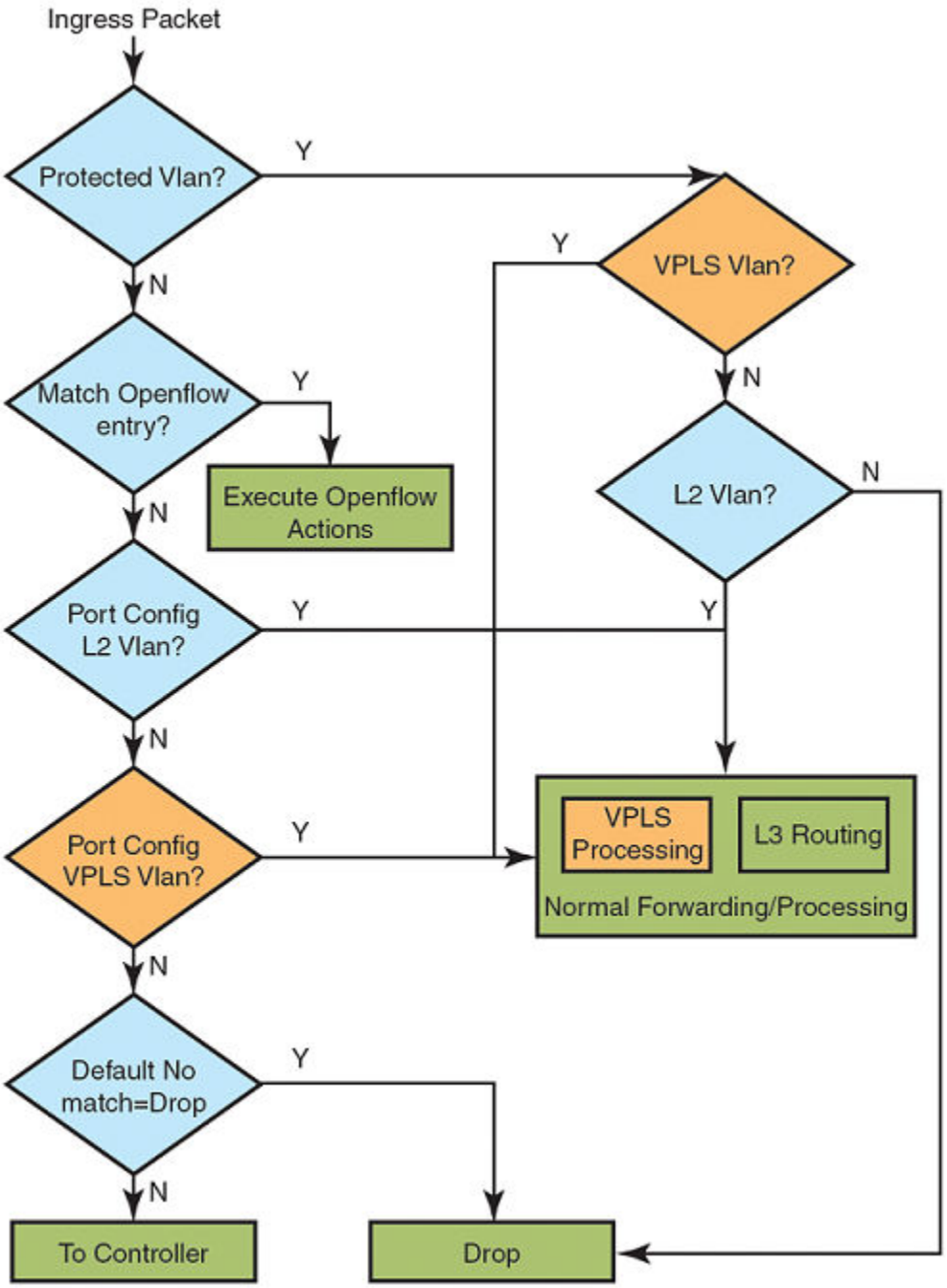
This feature supports VPLS switching or routing for protected VLANS and for configured unprotected VLANS on OpenFlow hybrid mode ports. It is supported on Layer 2, Layer 3 and Layer23 OpenFlow hybrid mode ports.

Limitations

- VPLS VLANs cannot be configured as Untagged VLANs on OpenFlow hybrid mode ports.
- OpenFlow unprotected VLANs for system maximum is limited to 4K. This number is utilized by both Layer 2 VLANs as well as VPLS VLANs.
- Layer 2 switching is not supported on OpenFlow hybrid mode ports.
- Dual tagged VPLS instances are not supported on OpenFlow hybrid mode ports.
- IGMP snooping is not supported on configured unprotected VLANs.

- If a VPLS instance is configured on an OpenFlow Layer 3 hybrid port and this VPLS VLAN is not protected (unprotected VLAN) then, in case of an incoming Layer 3 traffic (having same VLAN) hits a flow rule, it does not forward the traffic appropriately according to the flow rule.

FIGURE 4 Packet flow diagram



Sample configurations

VPLS support to configure OpenFlow hybrid mode port

For VPLS instance, you can configure a port as an OpenFlow hybrid mode port by executing these commands.

```
device(config)#
device(config)#router mpls
device(config-mpls)#
device(config-mpls)#vpls v1 100
device(config-mpls)#vpls-peer 17.17.17.17
device(config-mpls-vpls-v1)#vlan 100
device(config-mpls-vpls-v8-vlan-100)#tag e 2/8
device(config-mpls)#int e 2/8
device(config-if-e10000-2/8)#openflow enable layer2 hybrid-mode
```

Check for global **system-max** unprotected VLAN number, while configuring a port as a hybrid port. If the **system-max** unprotected VLAN number exceeds the maximum permissible number, the port is rejected from being configured as a hybrid mode port. Otherwise VPLS VLAN becomes the configured unprotected VLAN on the port.

VPLS support on configured protected VLANs

These are the steps to configure VLAN as an OpenFlow protected VLAN.

- Enable OpenFlow globally and configure **system-max** for the OpenFlow entries.
- Configure the OpenFlow Controller.
- Configure **system-max** for Protected VLAN entries.
- Enable OpenFlow hybrid on port.
- Configure protected VLAN for port.
- Configure VPLS instance on the port using the same configured protected VLAN.

To configure VLAN as an OpenFlow protected VLAN, execute the following commands.

```
device(config)#
device(config)#router mpls
device(config-mpls)#
device(config-mpls)#vpls v1 100
device(config-mpls)#vpls-peer 17.17.17.17
device(config-mpls-vpls-v1)#vlan 100
device(config-mpls-vpls-v8-vlan-100)#tag e 2/8
device(config-mpls)#int e 2/8
device(config-if-e10000-2/8)#openflow protected-vlans 100
```

NOTE

The maximum protected VLAN number is 40 on the port. When it exceeds, you get an error message, and the VLAN is not configured as a protected VLAN.

NOTE

If **system-max** protected VLANs exceeds 2K, then you get an error message, and the system does not configure the VLAN as protected VLAN.

VPLS instance to enable OpenFlow hybrid mode port on protected VLAN

Once protected VLANs have been configured, to enable the OpenFlow hybrid mode port, use the following commands to create a VPLS instance.

```
device(config)#
device(config)#router mpls
device(config-mpls)#
device(config-mpls)#vpls v1 100
device(config-mpls)#vpls-peer 17.17.17.17
```

```
device(config-mpls-vpls-v1)#vlan 100
device(config-mpls-vpls-v8-vlan-100)#tag e 2/8
device(config-mpls)#int e 2/8
device(config-if-e10000-2/8)#openflow protected-vlans 100
device(config-if-e10000-2/8)#openflow enable layer2 hybrid-mode
```

Since this protected VLAN has become part of VPLS VLAN on this port, VPLS switching on this protected VLAN is supported.

Deleting VPLS instances from protected VLAN

To remove VPLS instance for the protected VLAN from the OpenFlow hybrid mode port, use this sequence of commands.

```
device(config)#
device(config)#router mpls
device(config-mpls)#
device(config-mpls)#vpls v1 100
device(config-mpls)#vpls-peer 17.17.17.17
device(config-mpls-vpls-v1)#vlan 100
device(config-mpls-vpls-v8-vlan-100)#tag e 2/8
device(config-mpls)#int e 2/8
device(config-if-e10000-2/8)#openflow protected-vlans 100
device(config-if-e10000-2/8)#openflow enable layer2 hybrid-mode
device(config)#
device(config)#
device(config)#router mpls
device(config)#vpls v1 100
device(config)#vlan 100
device(config)#no tag e 2/8
```

Now this port is not a part of VPLS instance. The protected VLAN becomes unconfigured protected VLAN and It drops.

> NOTE
> When removing the protected VLAN configuration, do not exceed the **system-max** unprotected VLAN number. Otherwise the command is rejected and you get an error message.

After removing the protected VLAN configuration, this VPLS VLAN becomes configured as an unprotected VLAN. Now, It does VPLS switching in absence of matching flow.

Deleting VPLS instances from unprotected VLAN

To remove OpenFlow hybrid mode from a port with unconfigured protected VLAN and a VPLS instance, execute the following commands.

```
device(config)#
device(config)#router mpls
device(config-mpls)#
device(config-mpls)#vpls v1 100
device(config-mpls)#vpls-peer 17.17.17.17
device(config-mpls-vpls-v1)#vlan 100
device(config-mpls-vpls-v8-vlan-100)#tag e 2/8
device(config-mpls)#int e 2/8
device(config-if-e10000-2/8)#openflow protected-vlans 100
device(config-if-e10000-2/8)#openflow enable layer2 hybrid-mode
device(config)#
device(config)#router mpls
device(config)#vpls v1 100
device(config)#vlan 100
device(config)#no tag e 2/8
```

Now that, the port has become a normal port and VPLS instance is configured on the port, It does VPLS processing for that VLAN.

To remove OpenFlow hybrid mode from a port with configured unprotected VLAN and a VPLS instance, execute the following commands.

```
device(config)#
device(config)#router mpls
device(config-mpls)#
device(config-mpls)#vpls v1 100
device(config-mpls)#vpls-peer 17.17.17.17
device(config-mpls-vpls-v1)#vlan 100
device(config-mpls-vpls-v8-vlan-100)#tag e 2/8
device(config-mpls)#int e 2/8
device(config-if-e10000-2/8)#openflow enable layer2 hybrid-mode
device(config)#
device(config-if-e10000-2/8)#no openflow enable layer2 hybrid-mode
```

OpenFlow on individual LAG ports

This feature is to enable OpenFlow hybrid on LAG ports such that non OpenFlow traffic forwarding is also supported on a LAG port. The controller can configure OpenFlow and traditional flows on individual LAG ports.

These are the limitations, when configuring LAG ports.

- All LAG ports need to behave the same way for non OpenFlow traffic.
- You can configure OpenFlow only on the primary LAG port. Secondary LAG ports follow the same configuration as the primary LAG port.
- You cannot configure or delete OpenFlow on secondary LAG ports directly.
- When you look at the running configuration, the secondary ports is not shown even though the secondary LAG ports have OpenFlow configuration taken from the primary port.
- Secondary LAG ports take Protected VLANs configuration also from primary LAG port.
- OpenFlow enable or disable takes effect on secondary LAG port only when LAG is deployed.
- While LAG is deployed. LAG ports should not have any OpenFlow configuration. If they do, then LAG deployment fails. Similarly, while LAG is un-deployed, LAG ports should not have any OpenFlow configuration.

  NOTE
  This feature is applicable for MLX Series and XMR Series only.

This example shows before enabling OpenFlow on primary LAG port, Ports 1/3, 1/4 and 1/5 are LAG ports and 1/3 is the primary port. OpenFlow is enabled on 1/1 and 1/2.

```
device#show openflow interfaces
Total number of Openflow interfaces: 2

Port  Link     Speed Tag MAC            OF-portid   Name       Mode
1/1   Up       1G    Yes 000c.dbf5.bd00 1                      Layer2
1/2   Up       1G    Yes 000c.dbf5.bd01 2                      Layer2
```

After enabling OpenFlow on primary LAG port, enable OpenFlow on LAG primary port 1/3.

```
device (config-if-e1000-1/3)#openflow enable layer3 hybrid
device (config-if-e1000-1/3)#show openflow interfaces
Total number of Openflow interfaces: 5

Port  Link     Speed Tag MAC            OF-portid Name    Mode
1/1   Up       1G    Yes 000c.dbf5.bd00 1                 Layer2
1/2   Up       1G    Yes 000c.dbf5.bd01 2                 Layer2
1/3   Up       1G    Yes 000c.dbf5.bd01 3                 Hybrid-Layer3
1/4   Up       1G    Yes 000c.dbf5.bd01 4                 Hybrid-Layer3
1/5   Up       1G    Yes 000c.dbf5.bd01 5                 Hybrid-Layer3
```

To disable OpenFlow, delete OpenFlow configuration from the LAG primary port. This removes OpenFlow configuration from all secondary ports.

You must do the following to configure OpenFlow.

- Deploy LAG.
- Configure OpenFlow (and protected VLANS) on primary LAG port.
- Controller can configure flows on individual LAG ports.
- Unconfigure OpenFlow (and protected VLANS) on primary LAG port.
- Un-deploy LAG.

OpenFlow handles these LAG events.

- Deploy LAG: LAG ports should not have OpenFlow configuration at the time of LAG deployment. The system checks for OpenFlow configuration and rejects LAG deployment if any OpenFlow configuration exists.
- Un-deploy LAG: LAG ports should not have OpenFlow configuration at the time of LAG un-deployment. The system checks for OpenFlow configuration and rejects LAG un-deployment if any OpenFlow configuration exists on the primary LAG port.
- Enabling or disabling of OpenFlow on deployed primary LAG port: **OpenFlow enable or disable** done on deployed LAG primary port should be applied to all secondary LAG ports. If any of the LAG port has flows, **OpenFlow disable** fails. Similarly protected VLAN configuration done on primary LAG port should be applied to all secondary LAG ports.
- Addition or deletion of secondary port to deployed LAG: When you add secondary LAG port, OpenFlow configuration on primary port is applied to newly added port. Similarly, when you delete secondary port from LAG, OpenFlow configuration on that port is deleted.

Configuring OpenFlow

You can enable OpenFlow on an interface with Layer23 flows in order to support Layer 2 and Layer 3 flows on that interface. Layer23 flows support the OpenFlow hybrid port mode also. Configured with Layer23, the controller can configure flows with Layer 2 and Layer 3 parameters together. A flow can contain the following fields: Ingress port, Destination MAC address, Source MAC address, Ether type, VLAN ID , P-bits, Source IP address, Destination IP address, IP protocol, and IP DSCP.

By default, OpenFlow is disabled on devices. You must first enable OpenFlow on the device before you can configure the parameters on the device.

Enabling OpenFlow on devices

After you enable OpenFlow on the device, you can enable OpenFlow on specific interfaces and configure additional OpenFlow parameters.

Enabling OpenFlow on MLX Series and XMR Series devices

Enter the following command:

```
device(config)# openflow enable ofv100
```

The **ofv100** keyword specifies the OpenFlow protocol version supported.

Use the **no** form of the command to disable OpenFlow feature on the device.

NOTE
You must disable OpenFlow on all interfaces individually before you can disable OpenFlow globally on the device.

Enabling OpenFlow on CER 2000 Series and CES 2000 Series devices

You can specify the MAC address match rule capability as either source MAC or destination MAC address. Default is destination MAC address. On these devices, you cannot change the MAC address match option dynamically. You must first disable the current mode and then enable the new option. Changing the MAC address match option clears all existing OpenFlow configuration and OpenFlow flow table content.

Enter the following command:

```
device(config)# openflow enable ofv100
```

Use the **no** form of the command to disable OpenFlow feature on the device.

NOTE
You must disable OpenFlow on all interfaces individually before you can disable OpenFlow globally on the device.

Enabling OpenFlow on a specified interface

After you have enabled OpenFlow on the device, you can enable OpenFlow on specific interfaces.

NOTE
You can enable OpenFlow on an interface only after you have enabled OpenFlow globally on the device. In addition, you must use individual CLI commands to enable OpenFlow on each interface. You cannot specify a range of ports when enabling OpenFlow.

NOTE
Configuration of an OpenFlow hybrid port is not supported, if the port is already configured as a member of an MCT VLAN.

On CER 2000 Series and CES 2000 Series devices:

Enter the following command:

```
device(config-if-e1000-1/1)# openflow enable layer23 hybrid-mode
```

Use the **no** version of the command to disable OpenFlow on the interface. By default, the port is enabled for Layer 2 matching mode, since CER 2000 Series and CES 2000 Series devices currently do not support Layer 3 matching mode. Layer23 can be enabled with hybrid mode. If hybrid mode is enabled, that interface supports protected and unprotected VLANs similar to Layer 2 hybrid and Layer 3 hybrid.

On MLX Series and XMR Series devices

Enter the following command:

```
device(config-if-e1000-1/1)# openflow enable layer2
```

Or

```
device(config-if-e1000-1/1)# openflow enable layer3
device(config-if-e1000-1/1)# openflow enable layer23
```

You can specify Layer 2 or Layer 3 or both layers in hybrid mode as Layer23 matching mode to be supported on the interface. By default, interfaces on these devices support Layer 2 matching mode. If you enable Layer 2 matching mode on the specified interface, only Layer 2 matching fields are supported on that interface.

Flow validation

The following validations are required before programming flows on a Layer23 port:

- When IP fields exist in rule, then the Ether type must be 0x800.
- IPv6 rules are supported on the Layer23 port. (But IPv6 Ether type without IPv6 parameters is supported.)

Flow action

OpenFlow actions does not change for Layer23 support. All actions currently supporting Layer 2 or Layer 3 flows continue to be supported. Actions currently supported are listed separately for different devices.

On MLX Series and XMR Series devices:

When a matching flow entry is found, a set of actions can be applied for processing the packet. The system supports the following actions:

- Forward a packet to a port.
- Forward a packet to a set of ports.
- Forward a packet to a controller.
- Forward a packet received from a controller to a port or set of ports.
- Drop the packet.
- Keep, add, modify, or remove the VLAN ID or the VLAN priority. Modifying the VLAN ID per port is also supported (each destination port can send a packet with a different VLAN ID for the same matching rule).
- Modify the source MAC address and the destination MAC address for both Layer 2 and Layer 3 IPv4 flows.

On CER 2000 Series and CES 2000 Series devices:

When a matching flow entry is found, a set of actions can be applied for processing the packet. The system supports the following actions:

- Forward a packet to a port.
- Forward a packet to a set of ports.
- Forward a packet to a controller.
- Forward a packet received from a controller to a port or set of ports.
- Drop the packet.
- Keep, add, modify, or remove the VLAN ID or the VLAN priority. Modifying the VLAN ID per port is also supported (each destination port can send a packet with a different VLAN ID for the same matching rule).

Connecting to an OpenFlow Controller

To connect to an OpenFlow Controller in active mode, enter the following command:

```
device(config)# openflow controller ip-address 10.2.3.4
```

the IP address is the address of the OpenFlow Controller and SSL encryption is used by default. Also, the OpenFlow connection uses TCP port 6633.

To connect to an OpenFlow Controller in the passive mode, enter the following command:

```
device(config)# openflow controller passive no-ssl
```

You can optionally specify the TCP port to be used for the connection. By default, the device accepts the connection from a controller with any IP address. However, you can provide an IP address to limit which controller can connect to the device.

Use the **no** form of the command to remove a passive connection. Passive mode connections are intended for testing environments and not recommended for production environments.

Setting up SSL encryption for controller connections

By default, a connection to the controller uses SSL encryption. To set up SSL encryption, copy the SSL certificate and SSL client private key from the remote machine where you generated them into the device's flash using the following commands:

```
device(config)# copy tftp flash <remote ip> <remote file> client-certificate
device(config)# copy tftp flash <remote ip> <remote file> client-private-key
```

The IP address specifies the remote machine from which the SSL client certificate is being copied. The file name specifies the client certificate in the **copy tftp flash client-certificate** command, and the client private key in the **copy tftp flast client-private-key** command.

NOTE
SSL is not supported on passive controller connections.

The **remote file** variable specifies the file name of the client certificate in the first command, and the client private key in the second command.

For each controller, you must enter both the commands. The device can store up to three SSL certificates and client private keys. If you remove a controller connection, you must delete the SSL certificates and client private keys from the device's flash memory using the monitor mode commands.

NOTE
When using SSL to connect to the switch, the OpenFlow Controller can send only 50 flows at a time (for typical flows). This is not applicable to TCP-only connections to the OpenFlow controller.

NOTE
Refer *Extreme NetIron Management Guide* and *Extreme NetIron Monitoring Guide* for detailed information on setup.

Disabling an SSL client

You can disable the SSL client within the device using the following command:

```
device# ip ssl client disable
```

When you disable an SSL client in the device, the corresponding controller connection that used SSL encryption fails. However, you can re-enable the controller connection by removing the SSL encryption option from the controller connection. Use the **openflow controller ip-address no-ssl** command to disable SSL encryption in the connection.

Use the **no ip ssl client disable** command to re-enable the SSL client in the device.

Configuring multiple controller connections

These devices support up to three controller connections. You can configure these connections with active or passive modes, in any combination, such as all active, all passive, or some active and some passive. Each connection requires its own separate command. You can remove any of the connections using the **no** form of the **openflow controller ip-address** command. The following example shows how you configure three connections.

```
device(config)# openflow controller ip-address 10.2.3.4 no-ssl port 6635
device(config)# openflow controller ip-address 10.2.3.5 no-ssl
device(config)# openflow controller passive no-ssl ip-address 10.2.3.6
```

Configuring the system parameters for OpenFlow

You can specify the limit for OpenFlow flow table entries in the flow table using the following command:

```
device(config)# system-max openflow-flow-entries 304
```

You can specify the maximum number of flow table entries. The example shows 304 for the limit. The range is from 0 through 4096. The default number of flow table entries is zero.

> NOTE
> This command is not available on the CER 2000 Series and CES 2000 Series devices. The default number of flows table entries is 4096 on these devices.

Setting the system maximum

The **system-max openflow-pvlan-entries** command sets the CAM size of OpenFlow protected VLAN entries for the device. By default, this value is set to 0.

```
device(config)# system-max openflow-pvlan-entries 2000
```

The value represents the number of port and protected VLAN combination entries that can be configured in the system. The range is from 0 through 2048. The example shows 2000 for the value. After using this command, you must reload the system.

The **system-max openflow-unprotectedvlan-entries** command sets the CAM size of OpenFlow unprotected VLAN entries for the device. By default, this value is set to 0.

```
device(config)# system-max openflow-unprotectedvlan-entries 1000
```

The value represents the number of port and unprotected VLAN combination entries that can be configured in the system. The range is from 0 through 4096. After using this command, you must reload the system.

For Layer 2 or Layer 3 flows, use this command:

```
device(config)# system-max np-openflow-flow-entries layer2or3 1000 slot 1 to 2
```

For Layer 2 and Layer 3 IPv4 flows, use this command:

```
device(config)# system-max np-openflow-flow-entries layer23IPv4 1000
```

The following parameters are available for this command:

- layer2or3 -both Layer 2 flow or Layer 3 flow entries
- Layer23IPv4 - Layer23 including Layer 2 and IPv4 flow entries
- Layer23IPv6 - Layer23 including Layer 2, Layer 3 and IPv6 flow entries

- Layer3IPv6 - Layer 3 including IPv6 flow entries
- Slot number can be any of the valid slot number in the device. For slots, you can provide "all", "slot 1 to 2" and individual slot options.
- Slot number mentioned in a single command need not be of same card type.
- If any of the option layer2or3 or layer23IPv4 is not applicable for a particular card on a slot, it is ignored.
- You can have multiple lines of above command to have different sets of values for different set of modules.

NOTE
This command is only applicable for MLX Series and XMR Series devices.

NOTE
Always configure **system-max np-openflow-flow-entries** option values less than the configured number of flows in that particular slot.

Configuring the default action

By default, the device drops packets that do not match any of the programmed flows. However, you can configure a device-level option to forward the packets to the controller instead of dropping them. This is an optional configuration. If this option is not configured, packets that do not match any flow entries on a port are dropped. When sending a packet to the controller, a copy of the packet is sent to each of the configured controller connections.

To enable the default action, enter the following command:

```
device(config)# openflow default send-to-controller
```

Packets that match a flow entry on a port are processed according to the action specified and are not affected by this setting. Use the **no** form of the command to set the default action to drop such packets instead.

Displaying the OpenFlow status on the device

After enabling or disabling OpenFlow on a device, you can verify the configuration using any of the **show** commands.

Running configuration

When OpenFlow is enabled on the device, the **show running configuration** command displays output similar to the following:

```
device(config)# show running configuration
Current configuration:
ver V5.4.0iT183
mirror ethernet 1/19
openflow enable ofv100
```

Displaying the OpenFlow status

If OpenFlow is enabled on a device, you can get a detailed report of the OpenFlow status on that device.

CER 2000 Series and CES 2000 Series devices:

On these devices, only Layer 2 matching mode is supported on the interfaces.

```
device(config)# show openflow interface
```

```
Port Link  Port-State Speed Tag MAC            OF-portid Name       Mode
1/1  Up    Forward    1G    No  0000.00b4.89c1    1                 Layer2
1/2  Up    Forward    1G    No  0000.00b4.89c2    2                 Layer2
1/3  Up    Forward    1G    No  0000.00b4.89c3    3                 Layer2
```

MLX Series and XMR Series devices:

```
device(config)# show openflow interface
Port  Link Port-State Speed Tag MAC            OF-portid Name       Mode
1/5   Up   Forward    1G    No  0000.0088.0904    5      OpenFlow-A Layer2
1/7   Up   Forward    1G    No  0000.0088.0906    7      OpenFlow-B Layer3
1/8   Up   Forward    1G    No  0000.0088.0907    8      OpenFlow-C Layer3
2/1   Up   Forward    1G    Yes 0000.0088.0916    49     OpenFlow-E Layer2
```

TABLE 6 Output fields of the show openflow interface command

Field	Description
Port	Indicates the port number on the device.
Link	Indicates the link status.
Port-State	Indicates the action to be performed on packets that reach the interface. Supported actions include forward the packet, modify the VLAN tag, drop the packet, and send to the controllers.
Speed	Indicates the port speed.
Tag	Indicates whether the interface can accept tagged packets or not.
MAC	Indicates the MAC address of the port.
OF-PortID	Indicates the OpenFlow port ID that is assigned to the port on the device. Port numbers on the device are mapped to OpenFlow port IDs. For more information, see Behavior of ports and devices on page 37.
Name	Indicates the name assigned to the port.
Mode	Indicates the OpenFlow mode enabled on the port. Ports on CER 2000 Series and CES 2000 Series devices support only Layer 2 mode. Ports on MLX Series and XMR Series devices support either Layer 2 or Layer 3 mode.

In addition, you can use the **show interface** command at the interface level. If OpenFlow is enabled on the interface, the OpenFlow status is indicated in the output.

```
device(config-if-e1000-1/1)# show interfaces
GigabitEthernet1/1 is down, line protocol is down
  STP Root Guard is disabled, STP BPDU Guard is disabled
  Hardware is GigabitEthernet, address is 0000.0034.5060 (bia 0000.0034.5060)
  Configured speed auto, actual unknown, configured duplex fdx, actual unknown
  Member of VLAN 1 (untagged), port is in untagged mode, port state is Disabled
  STP configured to ON, Priority is level0, flow control enabled
  OpenFlow enabled, Openflow Index 1, Flow Type Layer2
  Priority force disabled, Drop precedence level 0, Drop precedence force disabled
  dhcp-snooping-trust configured to OFF
```

Displaying the configured connections to controllers

Use the **show openflow** command to display the OpenFlow configuration, including the configured connections to controllers on the device. The output includes the configured unprotected VLANs as well.

```
device(config)# show openflow
Administrative Status:       Enabled
SSL Status:                  Enabled
Controller Type:             OFV 100
Number of Controllers:       1
```

```
Controller 1:
Connection Mode:       passive, TCP,
Listening Address:     0.0.0.0
Connection Port:       6633
Connection Status:      TCP_LISTENING
Match Capability:
L2 : Port, Source MAC, Destination MAC, Ether type, Vlan, Vlan PCP
L3 : Port, Vlan, Vlan PCP, Ethertype(IP,ARP,LLDP), Source IP, Destination IP, IP Protocol, IP TOS, IP Src
Port, IP Dst Port
L23: All
Normal Openflow Enabled Ports:
Openflow Hybrid Interfaces:
e1/1
Protected VLANs   : None
Unprotected VLANs :     2, 3, 4, 5, 6, 7, 8, 9, 10, 11
..................................................
..................................................
3994, 3995, 3996, 3997, 3998, 3999, 4000, 4001, 4011,
e2/1
Protected VLANs   : None
Unprotected VLANs :  4010,
Default action: drop
Maximum number of flows allowed: 65536
Active flow: 0
Maximum number of Protected Vlans allowed: 2048
Maximum number of Unprotected Vlans allowed: 4096
Total number of Unprotected Vlans: 4002
```

TABLE 7 Output fields for the show openflow command

Field	Description
Administrative Status	Indicates the administrative status of OpenFlow on the device.
Controller Type	Indicates the OpenFlow protocol version that is supported on the device.
Number of Controllers	Lists the number of controller connections configured on the device. These devices support up to three concurrent controller connections.
Connection Mode	Indicates the mode of the controller connection configured. You can configure active or passive connection to controllers. An active connection is initiated by the device. In a passive connection, the device is in the listening mode, and accepts requests from controllers. If the optional controller address is not specified, any controller can establish a connection with the device in the passive mode. Refer to Connecting to an OpenFlow Controller on page 29.
Listening Address	Indicates the address of the specified controller.
Connection Port	Indicates the TCP port that is used for connection to the controller. By default, port 6633 is used.
Match Capability	Specifies the matching rules supported.
Normal OpenFlow Enabled Ports	Lists the ports on the device that are enabled for OpenFlow.
OpenFlow Hybrid Interfaces	Indicates the VLAN IDs.
Default action	Indicates the default action for packets that do not match any configured flows. By default, such packets are dropped. However, you can configure these packets to be sent to the controller by using the **openflow default send-to-controller** command.
Number of flows allowed	Indicates the maximum number of flows allowed on the device that is configured by using the **system-max openflow-flow-entries** command.

Displaying the data path ID of the device

OpenFlow associates a globally unique data path ID to be used by the controller to distinguish OpenFlow devices on a network. To display the data path ID assigned to the device, enter the following command:

```
device(config)# show openflow datapath-id

datapath-id# 0000001bedb3d0c0
```

The output of the command shows the data path ID. The data path ID is derived from the chassis MAC address.

Displaying the OpenFlow flows

You can display the OpenFlow flows that are configured on the device and their statistics by using the following command:

```
device(config)# show openflow flows eth 1/2
```

The **show openflow flows** command shows all the flows configured in the system flow table. If you specify the interface, all the flows configured in the system for that interface are displayed.

> NOTE
> On the CER 2000 Series and CES 2000 Series devices, statistics are available only for up to 2048 flows on a first-come-first-served basis.

```
device(config)# show openflow flows
Total Number of Flows:                                                  2
Total number of data packets sent to controller:      0
Total number of data bytes sent to controller:                                    0
Flow ID: 1 Priority: 32768 Status: Active
        Rule:
            In Port:         e1/2
            In Vlan:         Untagged
            Destination Mac: 000.0000.0001
            Destination Mac Mask:    FFFF.FFFF.FFFF
        Action: FORWARD
            Out Port: e1/1, Untagged
        Statistics: 0
            Total Pkts:  0
            Total Bytes: 0
Flow ID: 2 Priority: 32768 Status: Active
        Rule:
            In Port:                 e1/2
            In Vlan:                 Tagged [10]
            Vlan PCP:                 4
            Destinaltion Mac:     0000.0000.0001
            Destination Mac Mask:    FFFF.FFFF.FFFF
        Action: FORWARD
            Out Port: e1/1, Untagged
            Out Port: e1/3, Untagged
        Statistics: 0
            Total Pkts:  0
            Total Bytes: 0
```

TABLE 8 Output fields for the show openflow flows command

Field	Description
Port	Port ID
VLAN	VLAN ID
Flow ID	An identifier for each flow. You can use the flow ID from this output to display flow-specific details.

TABLE 8 Output fields for the show openflow flows command (continued)

Field	Description
Priority	The priority of the flow set by the controller when the flow is added, in the range 0 through 65536. If the priority value was not specified, the device assigns the default value, 32768.
Status	Indicates whether the flow is configured correctly in the device. An active status indicates a correctly configured flow.
Rule	Specifies the matching rule for the flow. In this example, the matching rule is for the flow with Flow ID 1, to forward untagged packets reaching the interface eth 1/2 with the destination MAC address of 0000.0000.0001, to the egress port eth 1/1 as untagged packets. Here, the destination MAC Address Mask of FFFF.FFFF.FFFF indicates that only packets exactly matching the specified destination MAC address are forwarded.
Statistics	Indicates the counter of packets and bytes. NOTE For CER 2000 Series and CES 2000 Series devices, only the number of packets are displayed.

Setting the OpenFlow purge timer

You can configure the maximum time before stale flows are purged from the OpenFlow flow table after a switchover, failover, or operating system upgrade.

The valid range is from 1 through 600. The default is 240 seconds.

You may not need to change the value of the OpenFlow purge timer under normal circumstances. If you anticipate a delay in learning the flows from the controller after switchover, you can configure a larger value for the OpenFlow purge timer.

The following example shows how to set the OpenFlow purge timer:

```
device(config)# openflow purge-time 500
device(config)# no openflow purge-time 350
```

The **no** form of this command sets the purge timer time to its default value.

Administrating OpenFlow

Clearing the OpenFlow statistics

You can clear the flow statistics for all flows or, optionally, for a specified flow. Only the counters of packets and bytes (when applicable) are cleared, none of the other flow table entries are affected.

To clear flow counters, enter the following command:

```
device(config)# clear statistics openflow
```

To clear flow counters for a specified flow, enter the following command:

```
device(config)# clear statistics openflow 2
```

In the example, the flow counters are cleared only for flow ID 2. Use the **show openflow flows** command to obtain flow IDs.

Deleting the OpenFlow flows

You can delete an individual OpenFlow rule or all the flows in the flow table. To delete a single OpenFlow rule based on a flow ID, enter the following command:

```
device# clear openflow flowid 6
device# clear openflow flowid all
```

The **clear openflow** command deletes the rule irrespective of the state it is in (ACTIVE, PENDING_ADD, PENDING_MODIFY, or PENDING_DELETE). The same rule can be added again later from the controller if needed.

Displaying OpenFlow tech-support

The **show tech-support openflow** command captures the output of multiple **show** commands at one time for diagnostic purposes.

```
device# show tech-support openflow
```

You can capture the output of the following commands:

- **show openflow datapath-id**
- **show openflow controller**
- **show openflow interface**
- **show openflow flows**
- **show versions**
- **show interfaces**
- **show statistics**
- **show running-config**
- **show logging**
- **show save**

OpenFlow configuration considerations

After you enable OpenFlow on a device, you can configure, generate, and monitor flows on the ports configured on the device from a controller on OpenFlow-enabled ports. The device flow table is entirely under the control of the OpenFlow Controller.

- OpenFlow action can duplicate traffic to 16 ports
- The OpenFlow Controller supports Administratively down (OFPPC_PORT_DOWN) through a Port Modification Message.

Behavior of ports and devices

- Ports that are enabled for OpenFlow cannot take part in any of the normal operations of the device, such as routing and Layer 2 forwarding. However, after OpenFlow is disabled on a port, the port can resume normal operations. This does not require disabling OpenFlow globally on the device.
- OpenFlow defines port numbers sequentially from 1. The `OF-portid` parameter in the output of the **show openflow interface** command is assigned to the ports on the device. On MLX Series and XMR Series devices, 48 OpenFlow ports are reserved per slot. OpenFlow port numbering starts from slot 1. That is, OpenFlow port 1 is port 1/1 (1/1 = slot 1/port 1), OpenFlow port 2 is port 1/2, and so on. Therefore, slot 1 has OpenFlow ports 1-48, slot 2 has OpenFlow ports 49-96, and so on.

For example, if slot 1 is an 8x10G card and slot 2 is an 8x10G card, then the OpenFlow ports are: slot 1 (OpenFlow ports 1 to 8); slot 2 (OpenFlow ports 49 to 56). On CER 2000 Series and CES 2000 Series devices, ports 1/1 to 1/48 are OpenFlow ports 1 to 48 and ports 2/1 and 2/2 are OpenFlow ports 49 and 50. The OpenFlow protocol offers a capability discovery message for the controller to discover the ports that are OpenFlow-enabled on the router and their capabilities.

- The flow table content is not cleared when the connection to a controller is lost. The device continues to forward traffic according to the flow entries defined in the flow table even in the absence of a controller connection.
- The flow table entries within the device are cleared when the device is reset.
- On the MLX Series and XMR Series devices, when the active management module in the device switches over, all controller connections are closed. Configured controller connections are re-established after the device switches over to the standby management module.
- Flow table entries associated with a port are maintained when a port goes down. When the port comes back up, those flow entries are restored on the port. Flow entries are removed only with an explicit command from the controller.
- When OpenFlow is disabled globally on the device using the **no openflow enable** command, the flow table in the device is cleared. However, before you can disable OpenFlow globally on the device, you must disable OpenFlow on all interfaces individually.
- When a controller tries to add a flow to the device with the same priority, rule, and action as a flow that exists in the flow table, the flow statistics are cleared (the system does not add a new flow). The following table summarizes the behavior for similar flows being successively added.

TABLE 9 Flow table behavior when flows similar to existing ones are added

Priority	Rule	Action	Device behavior
Same	Same	Same	Clear flow statistics
Same	Same	Different	– Update the action list – Clear the statistics
Same	Different	Same	Create new flow
Same	Different	Different	Create new flow
Different	Same	Same	Create new flow
Different	Same	Different	Create new flow

Removing an OpenFlow configuration from a device

In general, to remove OpenFlow from the device and make it a non-OpenFlow device, complete the following steps:

1. Disable OpenFlow on the ports where it is enabled.
2. Disable OpenFlow on the device globally.
3. (Optional) Set the maximum number of flows to zero using the **system-max openflow-flow-entries 0** command.
4. Reload the device.

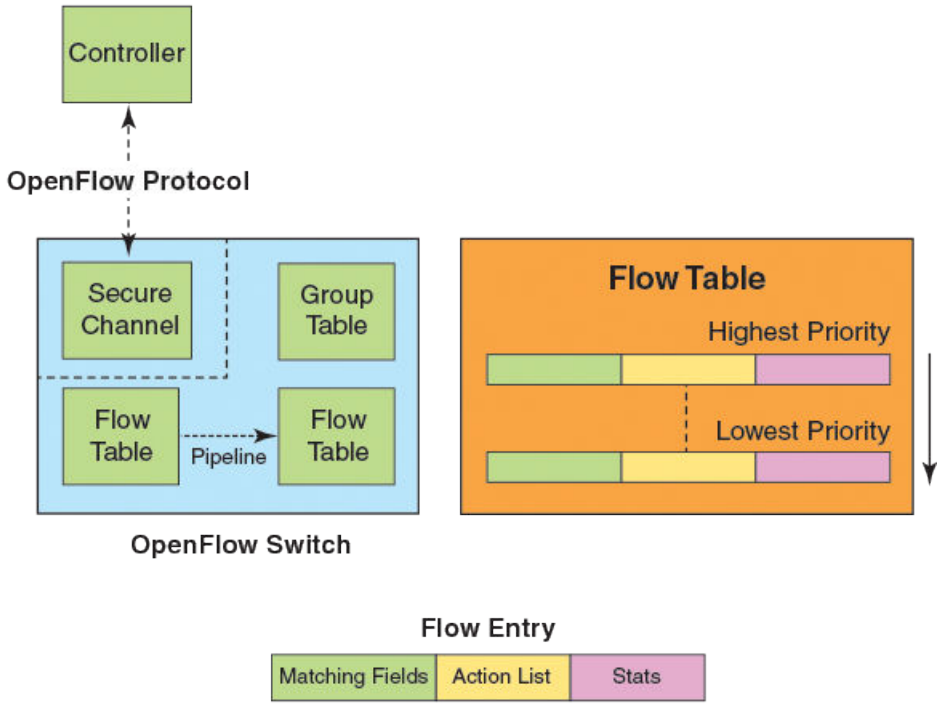
OpenFlow v1.3.0

Overview of OpenFlow v1.3.0

An OpenFlow switch maintains one or more flow tables, which are used for packet processing. The switch performs the actions listed in the table entry corresponding to the matched flow.

The OpenFlow Controller manages the OpenFlow switch using the OpenFlow. The OpenFlow Controller can add, delete, or modify flows by getting statistics for ports and flows and other information using the OpenFlow Protocol.

FIGURE 5 OpenFlow v1.3.0 architecture



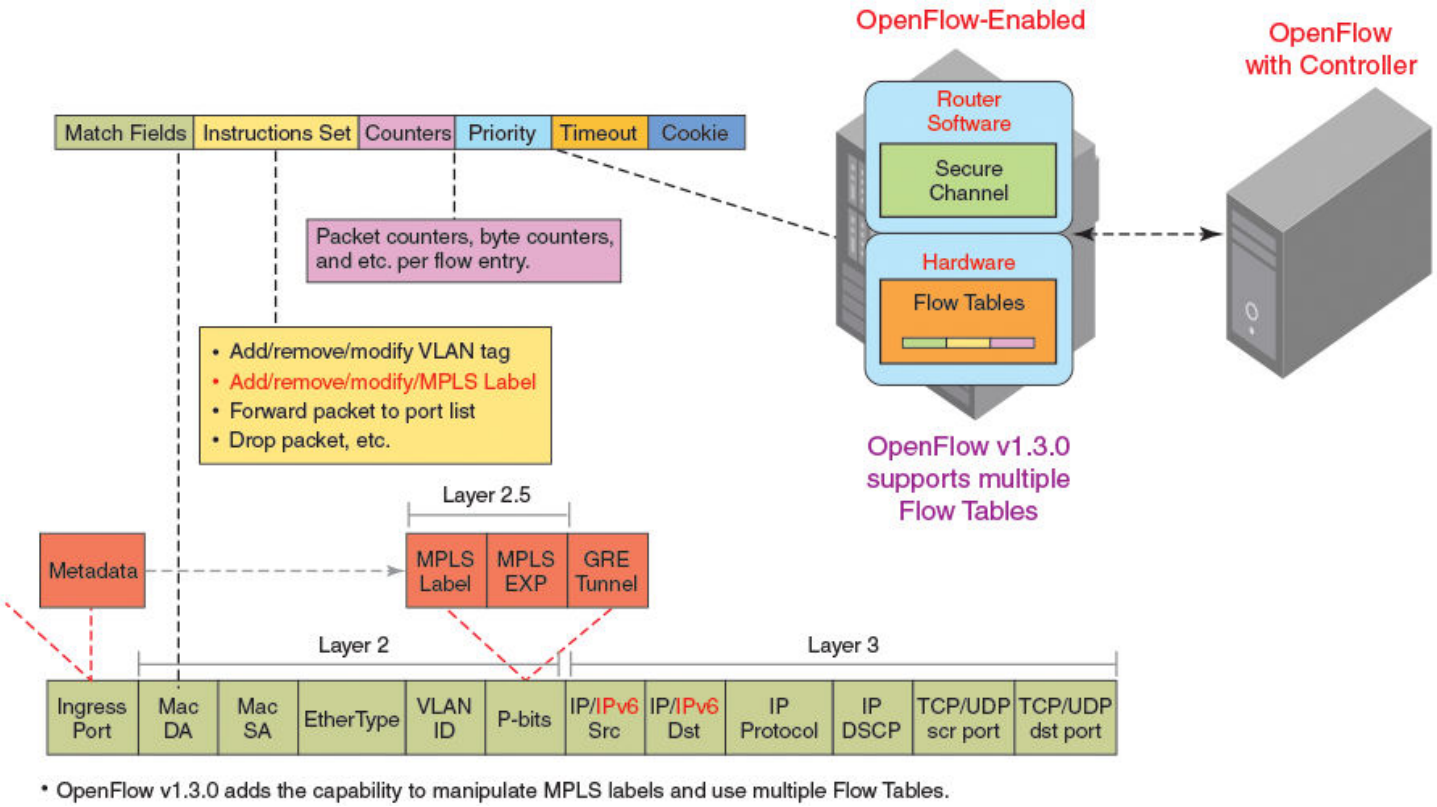
Each flow table maintained in a switch, consists of flow entries sorted by the flow priority. The highest priority flows are at the top of the flow table. Incoming packets are matched against the flow entries starting from the highest priority flow. If there is a match, then flow

matching stops, and the set of actions for that flow entry are performed. The packets that do not match any flow entry, are either dropped or sent to the controller.

OpenFlow v1.3.0 defines three types of tables:

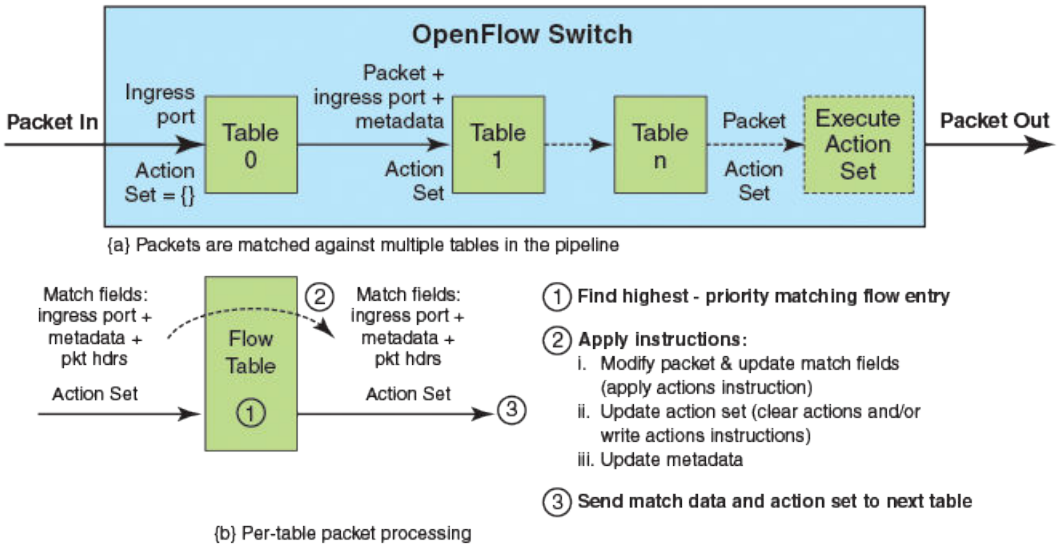
- Flow tables
- Group table
- Meter table

FIGURE 6 OpenFlow 1.3.0 flow table entries



The incoming packets are matched against the multiple tables in the pipeline.

FIGURE 7 Pipeline processing



Flow table entries

Each flow table entry contains the fields described in the following table.

TABLE 10 Flow table entries

Field	Description
Match fields	The match fields consist of ingress ports, packet header fields, and metadata from a previous flow table
Priority	Matching precedence of the entry
Counters	Statistics for matching packets
Instructions	Action set or pipeline processing
Cookie	Opaque data sent by the OpenFlow Controller

The following match fields are supported.

- All Layer 2 header fields
- All Layer 3 header fields

TABLE 11 OpenFlow match fields

Match field	MLX Series XMR Series	CER 2000 Series CES 2000 Series	Pre-requisite	Description
OXM_OF_IN_PORT	Yes	Yes	None	Ingress port. Numerical representation of incoming port, starting at 1. This may be a physical or switch-defined logical port.
OXM_OF_ETH_DST	Yes	Yes	None	Ethernet destination MAC address
OXM_OF_ETH_SRC	Yes	Yes	None	Ethernet source MAC address
OXM_OF_Ether type	Yes	Yes	None	Ethernet type of the OpenFlow packet payload, after VLAN tags.

TABLE 11 OpenFlow match fields (continued)

Match field	MLX Series XMR Series	CER 2000 Series CES 2000 Series	Pre-requisite	Description
OXM_OF_VLAN_VID	Yes	Yes	None	VLAN-ID 802.1Q header
OXM_OF_VLAN_PCP	Yes	Yes	None	VLAN-PCP 802.1Q header
OXM_OF_IP_DSCP	Yes	Yes	Ether type=0x0800	Diff Serv Code Point (DSCP). Part of the IPv4 TOS field or the IPv6 Traffic Class field.
OXM_OF_IP_PROTO	Yes IPv4 only	Yes IPv4 only	Ether type=0x0800	IPv4 or IPv6 protocol number
OXM_OF_IPV4_SRC	Yes	Yes	Ether type=0x0800	IPv4 source address. It can use subnet mask or arbitrary bit mask.
OXM_OF_IPV4_DST	Yes	Yes	Ether type=0x0800	IPv4 destination address. It can use subnet mask or arbitrary bit mask.
OXM_OF_TCP_SRC	Yes	Yes	IP PROTO = 6	TCP source port
OXM_OF_TCP_DST	Yes	Yes	IP PROTO = 6	TCP destination port
OXM_OF_UDP_SRC	Yes	Yes	IP PROTO = 17	UDP source port
OXM_OF_UDP_DST	Yes	Yes	IP PROTO = 17	UDP destination port

NOTE

IPv4 and IPv6 CAM must be allocated for Layer 23 ports if Ether type is not specified in the match for OXM_OF_ETH_DST & OXM_OF_ETH_SRC.

OpenFlow v1.3.0 instructions

Each flow entry has a set of instructions that are executed when the packet matches the entry.

The instruction set associated with each flow entry can have a maximum of one instruction of each type. The following table shows the actions supported on different devices.

TABLE 12 Actions for flow table instruction

Actions	Description	Support
Write-Action actions (Req)	Adds or overwrites specified actions to the action set.	Yes
Apply-Actions actions	Applies the specified actions immediately.	Yes
Clear-Actions actions	Clears all the actions in the action set.	Yes
Meter meter-id	Directs the packet to the specified meter.	Yes
Goto -Table next-table-id (Req)	Indicates the next table in pipeline processing.	No
Write-Metadata metadata/mask	Writes the metadata field from the mask.	No
Output (Req)	Forwards the packet to a specified OpenFlow port. If out-port is Controller, then the packet is sent as packet-in message.	Yes
Drop (Req)	No explicit drop action. Packet with empty action set is dropped.	Yes
Group	Processes the packet through the specified group.	Yes
Set field	Modifies the values of the packet header based on the field type.	Yes
Push-Tag/ Pop-Tag	Adds and removes tag (newly inserted tags are always the outermost tags).	Yes
Set-Queue	Enqueues the packet to a specific queue on the outgoing port.	Yes
Change TTL	Modifies the TTL value.	Yes

The set fields in the following table are supported for OpenFlow instructions. The set field action is used to set the value in the header field.

TABLE 13 Supported set field action

Set field	MLX Series XMR Series	CER 2000 Series CES 2000 Series	Description
OXM_OF_ETH_DST	Yes	No	Ethernet destination MAC address
OXM_OF_ETH_SRC	Yes	No	Ethernet source MAC address
OXM_OF_ETH_TYPE	No	No	Ether type of the OpenFlow packet payload after VLAN tags.
OXM_OF_VLAN_VID	Yes	Yes	VLAN-ID 802.1Q header
OXM_OF_VLAN_PCP	Yes	Yes	VLAN-PCP 802.1Q header
OXM_OF_IP_DSCP	Yes	Yes	Diff Serv Code Point (DSCP). Part of the IPv4 TOS field or the IPv6 Traffic Class field.

OpenFlow v1.3.0 actions

Each flow has a set of instructions that are executed when the packet matches the flow as per OpenFlow v1.3.0 specifications. Each flow can have a maximum of one instruction of each type.

A switch can reject a flow entry, if it is unable to execute the instructions associated with the flow entry. In this case, the switch returns an unsupported flow error. Flow tables may not support every match, every instruction, or every action.

TABLE 14 Instructions for OpenFlow actions

Instruction	Description
actions	Adds specified actions to the action set.
next-table-id	Indicates the next table in pipeline processing (One table is supported).
meter-id	Directs the packet to the specified meter.
apply-actions	Applies the specified actions immediately. The packet is modified and subsequent matching in the pipeline is done on the modified packet.
clear-actions	Clears all the actions in the action set.
write-metadata	Writes the metadata field from the mask.

These devices may support the actions listed in the following table.

TABLE 15 OpenFlow actions supported on different devices

OpenFlow action	MLX Series XMR Series	CER 2000 Series CES 2000 Series
Process the packet through the specified group	Yes	Yes
Add and remove tag	Yes	Yes
Add newly inserted tags always as the outermost tags	Yes	Yes
No explicit drop action. Packet with empty action set should be dropped.	Yes	Yes
Modify the values of the packet header based on the field type	Yes	Yes
Modify the TTL value	Yes	No

TABLE 15 OpenFlow actions supported on different devices (continued)

OpenFlow action	MLX Series XMR Series	CER 2000 Series CES 2000 Series
Set the queue ID for the packet	Yes	No

Multiple controller connections

An OpenFlow switch may be connected to multiple controllers for reliability, allowing the switch to continue to operate in OpenFlow mode if a controller or controller connection fails. The controllers coordinate the management of the switch amongst themselves to help synchronize controller handoffs.

Each controller can have one of the following roles:

- Equal: The controller has full access to the switch. It receives all the asynchronous messages from the switch and sends commands to modify the state of the switch (add or delete flows).
- Slave: The controller has a read-only access to the switch. It does not receive the asynchronous messages (apart from port status). It does not execute commands that modify the state of the switch: **packet-out, flow-mod, group-mod, port-mod**, or **table-mod**. The switch must reply with an OFPT_ERROR message, if it receives one of those commands from a Slave controller. Other controller-to-switch messages are processed normally.
- Master: The controller has full access to the switch as in the Equal role. When the controller changes its role to Master, the switch changes the other controller in the Master role to have the Slave role. The role change does not affect controllers with the Equal role.

A switch can be simultaneously connected to multiple controllers in the Equal role, multiple controllers in the Slave role, and, at most, one controller in the Master role. Each controller can communicate its role to the switch by way of an OFPT_ROLE_REQUEST message. This message can be used by the controller to set and query the role of its channel with the switch.

To detect the out-of-order messages during a Master-to-Slave transition, the OFPT_ROLE_REQUEST message contains a 64-bit generation ID, filed by sequence number, that identifies the mastership view. The controllers coordinate the assignment of generation IDs. The generation ID is a monotonically increasing counter. A new (larger) value is assigned each time the mastership view changes; that is, when a new Master is designated. The generation ID value wraps around once the maximum value has been reached.

```
device(config)# openflow controller
-------------------------------------------------------------------------
Contlr Mode  TCP/SSL IP-address     Port       Status            Role
-------------------------------------------------------------------------
1   (Equal)    passive  TCP    0.0.0.0          6633   TCP_LISTENING
2   (Master)   active   TCP    10.25.128.179    6633   OPENFLOW_ESABLISHED
3   (Slave)    active   TCP    10.25.128.177    6633   OPENFLOW_ESABLISHED
3   (Equal)    active   TCP    10.25.128.165    6633   OPENFLOW_ESABLISHED
```

Asynchronous configuration

Asynchronous messages may need to be sent to multiple controllers. An asynchronous message is duplicated for each eligible OpenFlow channel, and each message is sent when the respective controller connection allows it.

A controller can also control which types of switch asynchronous messages are sent over its OpenFlow channel. This is done using an asynchronous configuration message that has the filter setting for all the messages.

Different controllers can receive different notifications. A controller in the Master role can selectively disable notifications, and a controller in the Slave role can enable notifications it wants to monitor.

Each controller configuration block for active connection maintains its own asynchronous configuration setting for every role. The default initial configuration is shown in the following table.

TABLE 16 Action for asynchronous configuration

Messages	Bit field	Master or Equal role	Slave role
Packet-in reasons	No_match	Enable	Disable
	Action	Enable	Disable
	Invalid_TTL	Enable	Disable
Port status reasons	Add	Enable	Enable
	Delete	Enable	Enable
	Modify	Enable	Enable
Flow removed reasons	Idle_timeout	Enable	Disable
	Hard_timeout	Enable	Disable
	Delete	Enable	Disable
	Group_delete	Enable	Disable

NOTE

The asynchronous messages Action and Invalid_TTL are not supported by these devices. Controllers can set these bits in the filter setting and the device can accept the bits, but the messages are not sent out by the device.

Configuring source-interface for the controllers

This feature allows you to configure a source-interface to be used by the device for the connection to the controllers.

Currently, the devices support up to 3 active connections and 1 passive connection to the controllers. Controllers can be configured using the CLI interface of the routers. On configuring an active connection to a controller, the device tries and establishes a connection to the controller. However SYSLOG message is generated. The connection re-attempts every 15 seconds. If the given source-interface has multiple IP addresses configured, then the controller connection is initiated with the lowest IP address. However, the SYSLOG message is generated only for the first attempt to indicate the error.

- If the source-interface is down:

  The openflow source-interface used for the controller %I is down. Will retry the connection later.
- If there is no IP address configured:

  No IP address configured on the source-interface used for the controller %I. Will retry the connection later.
- When source interface is configured or unconfigured for a controller without force-reconnect option:

  Reset existing active connections to controller(s) for the new Source Interface configuration to take effect.

This feature is also applicable to active connections made using SSL. This feature is not supported on IPv6 interfaces.

To configure a source-interface for the connection from the device to the controller, enter the following command.

```
device(Config)#openflow controller source-interface ethernet 2/2
```

Supported OpenFlow messages

The following OpenFlow messages are supported on the devices.

TABLE 17 OpenFlow messages

Message type	MLX Series XMR Series	CER 2000 Series CES 2000 Series
OFPT_HELLO	Yes	Yes
OFPT_ERROR	Yes	Yes
OFPT_ECHO_REQUEST	Yes	Yes
OFPT_ECHO_REPLY	Yes	Yes
OFPT_EXPERIMENTER	No	No
OFPT_FEATURES_REQUEST	Yes	Yes
OFPT_FEATURES_REPLY	Yes	Yes
OFPT_GET_CONFIG_REQUEST	No	No
OFPT_GET_CONFIG_REPLY	No	No
OFPT_SET_CONFIG	No	No
OFPT_PACKET_IN	Yes	Yes
OFPT_FLOW_REMOVED	Yes	Yes
OFPT_PORT_STATUS	Yes	Yes
OFPT_PACKET_OUT	Yes	Yes
OFPT_FLOW_MOD	Yes	Yes
OFPT_GROUP_MOD	Yes	Yes
OFPT_PORT_MOD	No	No
OFPT_TABLE_MOD	No	No
OFPT_MULTIPART_REQUEST	Yes	Yes
OFPT_MULTIPART_REPLY	Yes	Yes
OFPT_BARRIER_REQUEST	Yes	Yes
OFPT_BARRIER_REPLY	Yes	Yes
OFPT_QUEUE_GET_CONFIG_REQUEST	Yes	No
OFPT_QUEUE_GET_CONFIG_REPLY	Yes	No
OFPT_ROLL_REQUEST	Yes	Yes
OFPT_ROLL_REPLY	Yes	Yes
OFPT_GET_ASYNC_REQUEST	Yes	Yes
OFPT_GET_ASYNC_REPLY	Yes	Yes
OFPT_SET_ASYNC	Yes	Yes
OFPT_METER_MOD	Yes	Yes

OpenFlow logical interface support

An OpenFlow switch supports three types of OpenFlow ports: physical ports, logical ports, and reserved ports. These standard ports can be used as output ports or for grouping. Port counters are associated with each of these ports.

The OpenFlow logical ports are switch-defined ports that do not correspond directly to a hardware interface of the switch. Logical ports are higher-level abstractions that may be defined in the switch using non-OpenFlow methods (for example, link aggregation groups, tunnels, loopback interfaces).

Logical ports may include packet encapsulation and may map to various physical ports. The processing done by the logical port must be transparent to OpenFlow processing and those ports must interact with OpenFlow processing like OpenFlow physical ports. The system assigns OpenFlow ID (OF ID) for physical ports upon enabling OpenFlow. The OF ID for each physical port takes the port number of the slot. For example, Port 1/1 (slot/port) is assigned OF ID 1, Port 1/2 is assigned OF ID 2, and so on.

After the last physical port has been assigned an OF ID, OF IDs are allocated to logical ports.

The difference between physical ports and logical ports is that a packet associated with a logical port may have an extra metadata field called Tunnel ID associated with it. When a packet received on a logical port is sent to the controller, both the logical port and its underlying physical port are reported to the controller. The controller pushes the flows using logical ports in flow rule as well as in flow action. An example is shown below.

```
dpctl tcp:10.37.226.145:7777 flow-mod cmd=add,table=0,idle=0,hard=0,prio=100
in_port=1060,eth_dst=00:c1:d1:00:01:22,eth_src=00:a1:b1:00:01:22,vlan_vid=689
apply:push_mpls=0x8847,set_field=mpls_label:434565,output=262145
```

Supporting IPsec tunnels as logical OpenFlow interfaces and groups

IPsec tunnels can be used as OpenFlow logical interfaces. The OpenFlow ID (OF ID) range is extended to enable configuration of IPsec logical interfaces (IPv4 and IPv6) and flows over these interfaces from the controller. The OF ID range is from 16385 to 24577. OpenFlow flow over IPsec logical interfaces contains rules to match IPv4 or IPv6 traffic (based on Layer 3 fields) with the IPsec interface OF ID as output action.

OpenFlow controller creates logical IPsec port groups, with a maximum of 8 IPsec ports per group. Only group type Select is supported. The configured flow over such groups contains rules to match IPv4 or IPv6 with IPsec group ID as output action. Flows over IPsec tunnel group are load-balanced across the individual tunnels that are part of the group.

The input port for the flows over IPsec port or IPsec port group can be Layer 3 or Layer23 hybrid and normal ports.

TABLE 18 Reserved OpenFlow IDs

Interface	Examples	Reserved range	Maximum OpenFlow ID
Physical Interface	Ethernet	1-1536	1536
Logical Interface	LSPs	262145-278529	16384
Logical Interface	IPsec	16385-24577	2048 **NOTE** This is the range within which a maximum of 2048 IPsec tunnel IDs can exist.

Packet-in and packet-out for IPsec logical interfaces

- Packet-out messages containing IPv4 or IPv6 data sent over OpenFow IPsec IPv4 or IPv6 tunnel interfaces respectively are supported. The packet in a packet-out message supports Layer 3 packet only. Pure Layer 2 packets are not supported as packet-out messages. Packet-out messages on OpenFlow IPsec groups are not supported.
- Packet-in messages into OpenFlow IPsec IPv4 or IPv6 tunnels are not supported.

Considerations and limitations

1. Generic flows (any input port) are supported with IPsec logical port as output action (but not with IPsec group as output action).
2. The logical OpenFlow IPsec interface is supported as a hybrid interface that supports both OpenFlow flows and regular flows.
3. The flow action involving OpenFlow IPsec logical port or a group can only have a single action that can either be an IPsec tunnel or an IPsec group as an output action. No other actions are supported.
4. OpenFlow IPsec logical interfaces or groups cannot be an input port for the flow.
5. Pure Layer 2 flows over IPsec ports or groups are not supported.
6. OpenFlow flows with matching IPv4 traffic rules can be sent over OpenFlow IPv4 IPsec tunnels only. Similarly, flows with matching IPv6 traffic rules can be sent over OpenFlow IPv6 IPsec tunnels only.
7. OpenFlow IPsec port groups can contain individual IPsec ports with the same IPsec mode (i.e., there cannot be an IPsec port group with IPv4 and IPv6 IPsec tunnel ports in the same group).
8. Heterogeneous grouping of logical ports is not supported, such as an IPsec logical group can only contain IPsec interfaces and no other OpenFlow logical interfaces like LSP interfaces.
9. Flows configured over logical IPsec ports, where the IPsec tunnels are created on LAG ports, are load-balanced on the individual LAG ports.

For all the CLI commands, please refer to *Extreme NetIron Command Reference*.

MPLS label-switched path as logical interface

The OpenFlow ID for MPLS label-switched paths (LSP's) is mapped one-to-one with its SNMP_TUNNEL_INDEX.

Port modification is supported for logical ports along with port status messages. By mapping OpenFlow to an MPLS LSP logical port, traffic follows OpenFlow rules for Layer 2 and Layer 3 matching fields and forwards the traffic to one or more MPLS LSP logical interfaces. For IPv4 over MPLS, it matches IPv4 traffic and forwards to the LSP logical interface. For IPv4-L3VPN, IPv6-L3VPN, and L2VPN, it pushes the L3VPN or VC label along for the MPLS LSP logical interface.

Limitations for MPLS LSP logical interfaces

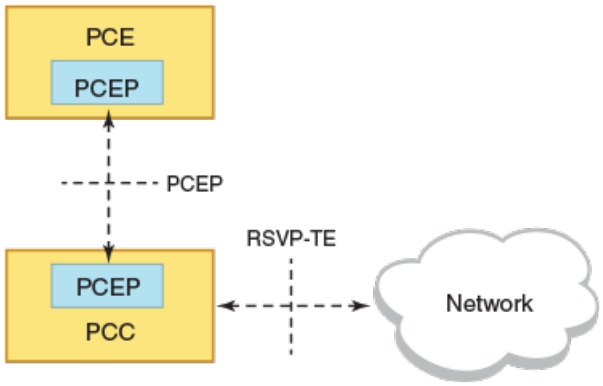
- MPLS LSP logical port supports VLAN modification (such as addition, modification or deletion action) as well as MPLS label, with L3VPN Label or VC Label using OpenFlow MPLS egress flow rule.
- An MPLS LSP Logical port is supported in OpenFlow groups.
- Static and bypass LSPs are not supported.
- Meter with DSCP REMARK is not supported.
- IPv6 over MPLS and IPv6VPN are not supported, but IPv6VPE is supported.
- An MPLS LSP logical port is not accepted as a match field in the OpenFlow rule.
- Any port validation that is applicable on physical ports, is applicable on logical ports as well.

- Pop VLAN action is supported by OpenFlow hybrid ports only as output in action.
- Multiple MPLS LSP OpenFlow logical ports or physical OpenFlow ports along with MPLS LSP OpenFlow logical interface ports are not supported together in the action.
- The maximum number of MPLS LSP tunnels is 1,000, and the maximum number of flows per tunnel is 512 pointing to the LSP tunnel.
- Generic input port and output as logical group are not supported.

Path Computation Element Protocol integration

The PCEP protocol is used by a Path Computation Client (PCC) to communicate with a Path Computation Element (PCE) as shown in the figure.

FIGURE 8 PCEP integration



Definitions:

- Path Computation Element (PCE) - an entity (component, application, or network node) capable of computing a network path or route based on a network graph and applying computational constraints.
- Path Computation Client (PCC) - any client application requesting a path computation to be performed by a PCE.
- Path Computation Element Protocol (PCEP) - TCP-based protocol to enable communications between a PCC and a PCE, or between two PCEs.

PCEP defines a set of messages and objects used to manage PCEP sessions and to request and send paths for multidomain traffic-engineered LSPs. PCEP provides a mechanism for a PCE to perform path computation for a PCC's external LSPs. The PCEP interactions include LSP status reports sent by the PCC to the PCE, and PCE updates for the external LSPs.

A TCP-based PCEP session connects a PCC to an external PCE. The PCC initiates the PCEP session and stays connected to the PCE for the duration of the PCEP session. During the PCEP session, the PCC requests LSP parameters from the stateful PCE. On receiving one or more LSP parameters from the PCE, the PCC re signals the TE LSP. When the PCEP session is terminated, the underlying TCP connection is closed immediately, and the PCC attempts to re-establish the PCEP session.

To integrate PCEP with OpenFlow, the same ID is used for the OpenFlow controller and the PCE. A router-configured LSP may get attributes from the PCE and can be used in OpenFlow flows when OpenFlow is enabled over the LSP.

Enabling OpenFlow over MPLS LSP

These commands allow the user to enable OpenFlow over individual MPLS LSP tunnels. It does not enable OpenFlow on all tunnels automatically though. These are only applicable on MLX Series and XMR Series.

```
Device(config-mpls-rsvp)#?
  backup-bw-best-effort         Provide bandwidth requested on backup path on best
                                effort basis
  clear                         Clear table/statistics/keys
  cls                           Clear screen
  delay-resv-send               Delay sending RESV for backup to come up before
                                protected session
  openflow-enable                       Enable openflow globally for rsvp tunnels
  end                           End Configuration level and go to Privileged level
  exit                          Exit current level
  no                            Undo/disable commands
  quit                          Exit to User level
  refresh-interval              Avg. interval between refresh path and resv msgs
  refresh-multiple              Num of unresponded path or resv before time out
  rsvp-hello                    Enable RSVP Hello on all mpls-interfaces
  rsvp-refresh-reduction        Enable RSVP Refresh Reduction Globally
  rsvp-reliable-messaging       Enable RSVP Reliable messaging globally
  show                          Display system information
  write                         Write running configuration to flash or
                                terminal
```

The following example shows the physical and the logical interfaces for **show openflow interfaces** command.

```
device # show openflow interfaces
Total number of Openflow Physical interfaces: 2

Port  Link     Speed Tag MAC            OF-portid   Name           Mode
1/1   Down     None  Yes 0024.38a4.1f00 1                          Layer2
1/2   Down     None  Yes 0024.38a4.1f01 2                          Layer2

Total number of Openflow Logical interfaces(Hybrid): 1
Interface-type  Oper-state  Name            Tunnel-ID   OF-portid
LSP             UP          abcd1         1           262145
```

To display enabled individual OpenFlow logical interfaces in detail, use the command **show openflow interface logical id**.

```
device#show openflow interface logical 262145
Interface Type : LSP
Name           : abcd1
Admin State    : DOWN
Oper State     : UP
Tx Pkt Count   : 0
Tx Byte Count  : 0
```

To display MPLS LSP in detail, use **show mpls lsp name** command.

```
device # show mpls lsp name abcd1
LSP abcd1, to 15.15.15.15, tunnel-interface index: 1
  From: 192.168.2.100, admin: UP, status: UP, tunnel interface(primary path): tnl1
  Times primary LSP goes up since enabled: 1
  Metric: 0
  Maximum retries: NONE, no. of retries: 0
  Pri. path: NONE, up: yes, active: yes
  Setup priority: 7, hold priority: 0
  Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
  CSPF-computation-mode configured: use te-metric(global)
  Constraint-based routing enabled: yes
   Path calculated using constraint-based routing: yes
   Path calculated using interface constraint: no
   Path calculated using te-metric
   Path cost: 1
  Tie breaking: random, hop limit: 0
  LDP tunneling enabled: no
  Soft preemption enabled: no
  OF-portid : 262145
```

```
  Active Path attributes:
   Tunnel interface: tnl1, outbound interface: e2/1
   Tunnel index: 2, Tunnel instance: 1 outbound label: 3
   Recorded routes:
    Protection codes/Rtr Id flag: P: Local  N: Node  B: Bandwidth  I: InUse R: RtrId
    192.168.20.1
```

To check OpenFlow is enabled over MPLS globally, use the following command.

```
device(config-mpls-rsvp)# openflow-enable
Error: Openflow is not enabled globally
```

To disable OpenFlow over the MPLS RSVP globally, use the following command.

```
device(config-mpls-rsvp)# no openflow-enable
Info: Openflow hybrid is already disabled over rsvp LSPs.
```

Enabling OpenFlow Hybrid over LSP

These commands enable OpenFlow hybrid over RSVP enabled LSP. After enabling OpenFlow , it generates an OF ID for it, that is known to the controller. The controller uses this ID for pushing the flows. These are only applicable on MLX Series and XMR Series as well.

```
device(config-mpls-lsp-tunnel10)#?
  adaptive                      enable lsp to be modified on fly
  auto-bandwidth                Enable auto-bandwidth on primary path
  bfd                           Set BFD options
  clear                         Clear table/statistics/keys
  cls                           Clear screen
  commit                        apply the parameter modifications to lsp
  cos                           Class of service
  cspf                          Enable or disable CSPF
  cspf-computation-mode         Set cspf computation mode for this LSP
  openflow-hybrid           Enable openflow hybrid on this LSP
  disable                       Tear down the LSP
  enable                        Establish the LSP
  end                           End Configuration level and go to Privileged level
  exclude-any                   Exclude any of the administrative groups
  exit                          Exit current level
```

To check OpenFlow is enabled over MPLS globally, use the following command.

```
device(config-mpls-lsp-abcd1)#openflow-hybrid
Error: Openflow is not enabled globally over RSVP Tunnels
```

If OpenFlow is already configured over the LSP, the following message comes up.

```
device(config-mpls-lsp-abcd1)#openflow-hybrid
Info: Openflow hybrid is already enabled over this LSP
```

OpenFlow IPv6 rule support

This feature enables support for matching IPv6 tuples for OpenFlow flows.

The match fields applicable for IPv6 traffic are shown below. OpenFlow supports matching of IPv6 tuples setup for these fields. When a match is found the corresponding action can be executed.

> NOTE
> The existing layer 2 match fields in each individual port mode are all supported for IPv6 match fields on MLX Series and XMR Series along with these match fields.

NOTE
IPv6 support on Layer23 port matches only the IPv6 Layer 3 header fields and VLAN tag on CER 2000 Series and CES 2000 Series.

TABLE 19 IPv6 OpenFlow match fields

Match field	MLX Series XMR Series	CER 2000 Series CES 2000 Series	Pre-requisite	Description
OFPXMT_OFB_IP_DSCP	Yes	Yes	Ether type=0x0800 or 0x86dd	IPv6 traffic class
OFPXMT_OFB_IP_ECN	No	No	Ether type=0x0800 or 0x86dd	IPv6 traffic class
OFPXMT_OFB_IP_PROTO	Yes	Yes	Ether type=0x0800 or 0x86dd	IPv6 next header
OFPXMT_OFB_TCP_SRC	Yes	Yes	IP PROTO = 6	TCP source port for IPv6 packets
OFPXMT_OFB_TCP_DST	Yes	Yes	IP PROTO = 6	TCP destination port for IPv6 packets
OFPXMT_OFB_UDP_SRC	Yes	Yes	IP PROTO = 17	UDP source port for IPv6 packets
OFPXMT_OFB_UDP_DST	Yes	Yes	IP PROTO = 17	UDP destination port for IPv6 packets
OFPXMT_OFB_IPV6_SRC	Yes	Yes	Ether type=0x86dd	IPv6 source address
OFPXMT_OFB_IPV6_DST	Yes	Yes	Ether type=0x86dd	IPv6 destination address
OFPXMT_OFB_IPV6_FLABEL	No	No	Ether type=0x86dd	IPv6 flow label
OFPXMT_OFB_ICMPV6_TYPE	Yes	Yes	IP PROTO = 58	IPv6 ND type
OFPXMT_OFB_ICMPV6_CODE	Yes	Yes	IP PROTO = 58	IPv6 ND code
OFPXMT_OFB_IPV6_ND_TARGET	No	No	ICMPv6 type=135 or 136	IPv6 ND field
OFPXMT_OFB_IPV6_ND_SLL	No	No	ICMPv6 type=135	IPv6 ND field
OFPXMT_OFB_IPV6_ND_TLL	No	No	ICMPv6 type=136	IPv6 ND field
OFPXMT_OFB_IPV6_EXTHDR	No	No	IP PROTO = 58	IPv6 extended header pseudo-field
OFPXMT_OFB_ICMPV4_TYPE	Yes	Yes	IP PROTO = 1	ICMPv4 type
OFPXMT_OFB_ICMPV4_CODE	Yes	Yes	IP PROTO = 1	ICMPv4 code

Actions

All flow actions for IPv4 matching flows are supported for IPv6 matching flows as well.

Match fields supported on CER 2000 Series and CES 2000 Series devices

The following table shows the supported rules on different types of OpenFlow ports.

TABLE 20 Match fields supported

Match field	Layer23	Layer 3
Source port	Yes	Yes
Destination MAC address	Yes[1]	No
Source MAC address	Yes[1]	No
VLAN ID	Yes	Yes

TABLE 20 Match fields supported (continued)

Match field	Layer23	Layer 3
VLAN PCP	Yes	Yes
Ether type	Yes	Yes
IPv4 SIP	Yes	Yes
IPv4 DIP	Yes	Yes
IPv6 SIP	Yes	Yes
IPv6 DIP	Yes	Yes
IP Protocol	Yes	Yes
L4 Source port	Yes	Yes
L4 Destination port	Yes	Yes
IP DSCP	Yes	Yes
IPv6 Traffic class	Yes	Yes
ICMPv6 Type	Yes	Yes
ICMPv6 Code	Yes	Yes
ICMPv4 Type	Yes	Yes
ICMPv4 Code	Yes	Yes

Limitations

- Ether type is compulsory in the OpenFlow rules for Layer 3 and Layer23 ports for CER 2000 Series and CES 2000 Series.
- IPv6 matching field is not supported for CER 2000 Series and CES 2000 Series. Any flow addition or modification with IPv6 matching field will be rejected when the input port is configured in OpenFlow Layer 2 mode.

IPv6 packets

IPv6 packets consist of IPv6 header and upper layer Protocol Data Unit (PDU) and sometimes extension headers as well. The size of the IPv6 header is 40 bytes. The fields in the IPv6 header are: Source IP, Destination IP, Next Header (IP Protocol), Traffic class, Flow label, Hop limit. A Next Header field in the IPv6 header indicates the next extension header. IPv6 extension headers have no maximum size and can expand to accommodate all the extension data needed for IPv6 communication.

The upper layer Protocol Data Unit (PDU) usually consists of an upper layer protocol header and its payload, for example an ICMPv6 message, a UDP message, or a TCP segment.

This table shows the typical value of the Next Header field for an IPv6 header or an IPv6 extension header.

TABLE 21 Next Header fields

Value (in decimal)	Header
0	Hop-by-hop options header
6	TCP
17	UDP
41	Encapsulated IPv6 header
43	Routing header
44	Fragment header
46	Resource Reservation Protocol (RRP)
50	Encapsulating security payload
51	Authentication header

TABLE 21 Next Header fields (continued)

Value (in decimal)	Header
58	ICMPv6
59	No next header
60	Destination options header

Scalability for IPv6

To support IPv6 matching, hardware entries are reserved in IPv6 session and IPv6 SuperACL partitions.

System-max command **np-openflow-entries** is modified to support IPv6.

Scaling for MLX Series and XMR Series devices

The following parameters are available for this command:

- layer3IPv6-To add IPv6 matching flows on OpenFlow layer 3 ports
- layer23IPv6-To add IPv6 matching (along with Layer 2 fields) flows on OpenFlow layer23 ports

NOTE
IPv6 SuperACL is not supported on Gen1 cards and Interface modules. So layer23IPv6 command is rejected on these devices.

NOTE
layer2or3, layer23IPv4, layer3IPv6 and layer23IPv6 configurations are independent of each other and should be in separate lines for configuration generations.

TABLE 22 System-max matching fields

System-max	Matching fields
layer2or3	Layer 2 or Layer 3 fields
layer23IPv4	Layer 2 and Layer 3 fields (non IPv6 traffic)
layer3IPv6	Layer 3 fields for IPv6 traffic
layer23IPv6	Layer 2 and Layer 3 fields for IPv6 traffic

```
system-max np-openflow-entries {layer2or3 | layer23ipv4 | layer3ipv6 | layer23ipv6} slot [i j k | <i to z>
| <all>]
```

Scaling for CER 2000 Series and CES 2000 Series devices

TABLE 23 System-max matching fields

Device	Layer 2	Layer23	Layer 3
CES series[1]	4k	2k	2k
CER series[2]	32k	2k	2k

OpenFlow interface support on hybrid and non-hybrid ports

These are the rules supported on MLX Series and XMR Series for both hybrid and non-hybrid OpenFlow ports.

For OpenFlow Layer 2 ports:

- OpenFlow Layer 2 interfaces can not have flows matching IPv6 fields.

For OpenFlow Layer 3 ports:

- OpenFlow Layer 3 interfaces can have flows matching IPv6 fields similar to IPv4 fields. OpenFlow Layer 3 interface expects Ether type in flow matching. So, a flow can match one of IPv4, IPv6 or non-IP (LLDP, ARP).

For OpenFlow Layer23 ports:

- OpenFlow Layer23 interfaces can have flows matching IPv6 fields similar to IPv4 fields. OpenFlow Layer23 interface can have flows matching Layer 2 fields along with IPv6 fields.

  NOTE
  Flow matching of IPv6 fields is not supported on Layer23 ports on CER 2000 Series and CES 2000 Series devices.

  NOTE
  CER 2000 Series and CES 2000 Series devices do not support hybrid port mode on OpenFlow ports.

Displaying the OpenFlow flows

You can display the OpenFlow flows that are configured on the devices with IPv6 fields. The **show openflow flows** command shows all the flows configured in the system flow table. If you specify the interface, all the flows configured in the system for that interface are displayed.

Below is an example to show the IPv6 fields for both in MP and LP. Enter the following command:

```
device(config)# show openflow flows
Flow ID: 8 Priority: 32768 Status: Active
        Rule:
           In Port:      generic
           Ether type:   0x000086dd
           IP Protocol:          58
           Source IPv6:          2001:db8:85a3:42:0:8a2e:370:7334/128
           Destination IPv6:     2001:2:6c::430/48
           ICMPV6 Type:          1
           ICMPV6 Code:          3
        Instructions: Apply-Actions
                Action: FORWARD
                        Out Port:  e1/15
        Statistics:
           Total Bytes: 0
```

show openflow flows have ICMPv4 fields in match for both in MP and LP as following.

```
device(config)# show openflow flows
Flow ID: 6 Priority: 32768 Status: Active
        Rule:
           In Port:      generic
           Ether type:   0x00000800
           Source IP:            51.1.1.1       Subnet IP:       255.255.255.255
           Destination IP:       40.1.1.1       Subnet IP:       255.255.255.0
           IP Protocol:          1
           ICMPV4 Type:          3
           ICMPV4 Code:          2
        Instructions: Apply-Actions
                Action: FORWARD
                        Out Port:  e1/16
        Statistics:
           Total Bytes: 0
```

The **show openflow flows** command has the match capability as below.

Layer 2: Port, Source MAC address, Destination MAC address, Ether type, VLAN, VLAN PCP

Layer 3 : Port, VLAN, VLAN PCP, Ether type (IP, IPv6 ,ARP and LLDP), Source IP address, Destination IP address, IP Protocol, IP TOS , IP Source Port, IP Destination Port, Source IPv6 address, Destination IPv6 address, ICMPv6 Type, ICMPv6 code ICMPv4 Type, ICMPv4 code

Layer23: All

NOTE
IP Protocol represents both IPv4 and IPv6 protocol numbers depending on the matching rules for IPv4 or IPv6 fields.

NOTE
Matching IP TOS supports both IPv4 DSCP and IPv6 traffic class.

CAM partition for MLX Series and XMR Series

This section indicates maximum available IPv6 Session and IPv6 SuperACL CAM space for OpenFlow.

Total available CAM space is divided into following parts.

- OpenFlow flows
- OpenFlow protected VLANs
- OpenFlow unprotected VLANs (only for Interface modules)

These are the rules followed for the CAM partition for OpenFlow.

1. IPv6 rule ACL need 128 entries by default.
2. CAM sub-partition is not needed for unprotected VLANs except for Interface modules.
3. **ipv6-mcast-cam** system-max value is set to 0 to maximize IPv6 Session and IPv6 SuperACL CAM usage. Default value for this system-max is 2048.
4. IPv6 SuperACL use CAM space from IPv6 Session. Default value for this system-max is 0.

   NOTE
   For detailed information on CAM partition for IPv6 Session and IPv6 SuperACL, refer *Extreme NetIron Management Guide* and *Extreme NetIron Monitoring Guide*.

CAM partition for OpenFlow

CAM is partitioned on the device by a variety of profiles that you can select depending on your application. The available profiles for XMR Series and MLX Series are listed in *Extreme NetIron Management Guide* and *Extreme NetIron Monitoring Guide*.

To implement a CAM partition profile, enter the following command.

```
device(config) cam-partition profile ipv4
```

Syntax: cam-partition profile [**ipv4** | **ipv4-ipv6** | **ipv4-ipv6-2** | **ipv4-vpls** | **ipv4-vpn** | **ipv6** | **l2-metro** | **l2-metro-2** | **mpls-l3vpn** | **mpls-l3vpn-2** | **mpls-vpls** | **mpls-vpls-2** | **mpls-vpn-vpls** | **multi-service** | **multi-service-2** | **multi-service-3** | **multi-service-4**]

The *ipv4* parameter adjusts the CAM partitions, for XMR Series and for MLX Series, to optimize the device for IPv4 applications.

The *ipv4-ipv6* parameter adjusts the CAM partitions, for XMR Series and for MLX Series, to optimize the device for IPv4 and IPv6 dual stack applications.

The Openflow CAM partition profiles for IPv6 ACL session for Generation 1 and Generation 2 modules are displayed below.

TABLE 24 Generation 1 CAM partitioning profiles available for MLX Series routers, XMR Series routers, and BR-MLX-10GX24-DM modules

Profile	IPv6 ACL MLX Series routers	IPv6 ACL XMR Series routers	IPv6 ACL BR-MLX-10GX24-DM modules **NOTE** 180 bits per CAM are supported. Two CAMs equal 1 IPv6 session CAM.
Default Profile	2K	4K	16K
ipv4 Profile	0	0	0
ipv6 Profile	12K	24K	32K
l2-metro Profile	0	0	0
l2-metro-2 Profile	0	0	0
multi-service Profile	4K	8K	16K
multi-service-2 Profile	2K	4K	8K
multi-service-3 Profile	4K	8K	16K
multi-service-4 Profile	4K	8K	16K
multi-service-5 Profile	4K	8K	8K
mpls-vpn-vpls Profile	0	0	0
mpls-l3vpn Profile	0	0	0
mpls-l3vpn-2 Profile	0	0	0
mpls-vpls Profile	0	0	0
mpls-vpls-2 Profile	0	0	0
ipv4-vpn Profile	0	0	0
ipv4-ipv6 Profile	10K	20K	8K
ipv4-vpls Profile	0	0	0
ipv4-ipv6-2Profile	4K	8K	16K
Entry size (in bits)	640	640	360

TABLE 25 Generation 2 CAM partitioning profiles for IPv6 ACL are displayed in the columns below for each module

Profile	IPv6 ACL BR-MLX-10GX8-M	IPv6 ACL BR-MLX-10GX8-X	IPv6 ACL NI-MLX-10GX8-D	IPv6 ACL/Multicast IPv6 ACL BR-MLX-100GX1-X(4) BR-MLX-100GX2-X(4)	IPv6 ACL BR-MLX-40GX4-M	IPv6 ACL BR-MLX-10GX20-M (1G/10G combo), BR-MLX-1GX20-U10G-M and BR-MLX-100GX2-CFP2-M	IPv6 ACL BR-MLX-10GX20-X2 (1G/10G combo), BR-MLX-1GX20-U10G-X2 and BR-MLX-100GX2-CFP2-X2
Default Profile	2K	4K	2K	16K	6K	4K	12K
ipv4 Profile	0	0	0	0	0	0	4K
ipv6 Profile	12K	24K	12K	48K	20K	24K	60K
l2-metro Profile	0	0	0	0	0	0	0

TABLE 25 Generation 2 CAM partitioning profiles for IPv6 ACL are displayed in the columns below for each module (continued)

Profile	IPv6 ACL BR-MLX-10GX8-M	IPv6 ACL BR-MLX-10GX8-X	IPv6 ACL NI-MLX-10GX8-D	IPv6 ACL/Multicast IPv6 ACL BR-MLX-100GX1-X(4) BR-MLX-100GX2-X(4)	IPv6 ACL BR-MLX-40GX4-M	IPv6 ACL BR-MLX-10GX20-M (1G/10G combo), BR-MLX-1GX20-U10G-M and BR-MLX-100GX2-CFP2-M	IPv6 ACL BR-MLX-10GX20-X2 (1G/10G combo), BR-MLX-1GX20-U10G-X2 and BR-MLX-100GX2-CFP2-X2
l2-metro-2 Profile	0	0	0	0	0	0	0
multi-service Profile	4K	8K	4K	32K	6K	8K	20K
multi-service-2 Profile	2K	4K	2K	16K	8K	4K	12K
multi-service-3 Profile	4K	8K	4K	24K	8K	8K	20K
multi-service-4 Profile	4K	8K	4K	32K	8K	8K	24K
multi-service-5 Profile	4K	8K	8K	16K	4K	8K	28K
multi-service-6 Profile	12K	12K	4K	32K	8K	8K	24K
telemetry-1 Profile	12K	12K	6K	24K	12K	20K	28K
mpls-vpn-vpls Profile	0	0	0	0	0	0	0
mpls-l3vpn Profile	0	0	0	0	0	0	0
mpls-l3vpn-2 Profile	0	0	0	0	0	0	0
mpls-vpls Profile	0	0	0	0	0	0	0
mpls-vpls-2 Profile	0	0	0	0	0	0	0
ipv4-vpn Profile	0	0	0	0	0	0	0
ipv4-ipv6 Profile	10K	20K	10K	40K	18K	20K	36
ipv4-vpls Profile	0	0	0	0	0	0	0
ipv4-ipv6-2Profile	4K	8K	4K	16K	8K	8K	4

TABLE 25 Generation 2 CAM partitioning profiles for IPv6 ACL are displayed in the columns below for each module (continued)

Profile	IPv6 ACL BR-MLX-10GX8-M	IPv6 ACL BR-MLX-10GX8-X	IPv6 ACL NI-MLX-10GX8-D	IPv6 ACL/Multicast IPv6 ACL BR-MLX-100GX1-X(4) BR-MLX-100GX2-X(4)	IPv6 ACL BR-MLX-40GX4-M	IPv6 ACL BR-MLX-10GX20-M (1G/10G combo), BR-MLX-1GX20-U10G-M and BR-MLX-100GX2-CFP2-M	IPv6 ACL BR-MLX-10GX20-X2 (1G/10G combo), BR-MLX-1GX20-U10G-X2 and BR-MLX-100GX2-CFP2-X2
Entry size (in bits)	640	640	640	320	640	640	640

TABLE 26 CAM partitioning profiles for IPv6 SuperACL (shared with IPv6 ACL) are displayed in the columns below for each module

Profile	IPv6 SuperACL (shared with IPv6 ACL BR-MLX-10GX8-M	IPv6 SuperACL (shared with IPv6 ACL BR-MLX-10GX8-X	IPv6 SuperACL (shared with IPv6 ACL NI-MLX-10GX8-D	IPv6 SuperACL (shared with IPv6 ACL BR-MLX-100GX1-X(4) BR-MLX-100GX2-X(4)	IPv6 SuperACL (shared with IPv6 ACL BR-MLX-40GX4-M	IPv6 SuperACL (shared with IPv6 ACL BR-MLX-10GX20-M (1G/10G combo), BR-MLX-1GX20-U10G-M and BR-MLX-100GX2-CFP2-M BR-MLX-10GX20-X2 (1G/10G combo), BR-MLX-1GX20-U10G-X2 and BR-MLX-100GX2-CFP2-X2
Default Profile	0K	0 - 2K	0K	0 - 6K	0- 4K	0- 2K
ipv4 Profile	0	0	0	0	0	0
ipv6 Profile	0-10K	0-22K	0-10K	0-22K	0 - 18K	0-22K
l2-metro Profile	0	0	0	0	0	0
l2-metro-2 Profile	0	0	0	0	0	0
multi-service Profile	0 - 2K	0 - 6K	0 - 2K	0 -14K	0 -4K	0 - 6K
multi-service-2 Profile	0	0 -2K	0	0 -6K	0 -6K	0 -2K
multi-service-3 Profile	0 -2K	0 -6K	0 -2K	0 -10K	0 -6K	0 -6K
multi-service-4 Profile	0 -2K	0 -6K	0 -2K	0 -14K	0 -6K	0 -6K
multi-service-5 Profile	0 - 2K	0 - 6K	0 - 6K	0 -6K	0 -2K	0 - 6K

TABLE 26 CAM partitioning profiles for IPv6 SuperACL (shared with IPv6 ACL) are displayed in the columns below for each module (continued)

Profile	IPv6 SuperACL (shared with IPv6 ACL BR-MLX-10GX8-M	IPv6 SuperACL (shared with IPv6 ACL BR-MLX-10GX8-X	IPv6 SuperACL (shared with IPv6 ACL NI-MLX-10GX8-D	IPv6 SuperACL (shared with IPv6 ACL BR-MLX-100GX1-X(4) BR-MLX-100GX2-X(4)	IPv6 SuperACL (shared with IPv6 ACL BR-MLX-40GX4-M	IPv6 SuperACL (shared with IPv6 ACL BR-MLX-10GX20-M (1G/10G combo), BR-MLX-1GX20-U10G-M and BR-MLX-100GX2-CFP2-M BR-MLX-10GX20-X2 (1G/10G combo), BR-MLX-1GX20-U10G-X2 and BR-MLX-100GX2-CFP2-X2
mpls-vpn-vpls Profile	0	0	0	0	0	0
mpls-l3vpn Profile	0	0	0	0	0	0
mpls-l3vpn-2 Profile	0	0	0	0	0	0
mpls-vpls Profile	0	0	0	0	0	0
mpls-vpls-2 Profile	0	0	0	0	0	0
ipv4-vpn Profile	0	0	0	0	0	0
ipv4-ipv6 Profile	0 - 8K	0 -18K	0 -8K	0 -18K	0 -16K	0 -18K
ipv4-vpls Profile	0	0	0	0	0	0
ipv4-ipv6-2Profile	0 -2K	0 -6K	0 -2K	0 -6K	0 -6K	0 -6K
Entry size (in bits)	640	640	640	640	640	640

Supporting untagged VLAN on protected and configured unprotected VLAN

Devices support untagged traffic on protected VLAN or configured unprotected VLAN supporting IP traffic on OpenFlow hybrid port. You can configure untagged VLAN as protected VLAN or unprotected VLAN.

This feature supports untagged traffic on OpenFlow hybrid port to be treated as protected VLAN or configured unprotected VLAN traffic and such traffic is subjected to IP forwarding or routing.

- When OpenFlow is enabled on a port (hybrid or non-hybrid), the port is moved from untagged member of default VLAN (say VLAN 1) to untagged member of 4095 internally thus it is not displayed as untagged in 4095.
- Port default VLAN is always set to 4095 and is able to send untagged packets (not matching flow) to CPU.

- Untagged traffic is treated as OpenFlow traffic.
- OpenFlow port cannot become untagged member of any VLAN other than default VLAN.

Configuring VLANs

Assuming the port is configured as an OpenFlow hybrid port, the following are the configuration options to consider.

Case 1: When a port is added as untagged in unprotected VLAN making the port configured unprotected VLAN.

Configuration is accepted and untagged traffic on the port 2/1, for example, is forwarded as per the OpenFlow rule if matching rule is present. If Matching OpenFlow rule is not present, untagged traffic on port 2/1 is routed as per routing table. If route is not present, untagged traffic is processed according to the default OpenFlow rule (drop or send to controller).

```
device(config-if-e10000-2/1)#openflow enable layer2 hybrid-mode
device(config-if-e10000-2/1)#vlan 300
device(config-vlan-300)#untagged ethernet 2/1
```

Case 2: When a port is added as untagged in a protected VLAN.

Configuration is accepted and untagged traffic on port 2/1, for example, is forwarded as per route table.

```
device(config-if-e10000-2/1)#openflow enable layer2 hybrid-mode
device(config-if-e10000-2/1)#openflow protected-vlans 400
device(config-if-e10000-2/1)#vlan 400
device(config-vlan-400)#untagged ethernet 2/1
```

Case 3: When a port is removed from configured unprotected VLAN.

Configuration is accepted and untagged traffic on port 2/1 is forwarded as per OpenFlow rule, if matching rule is present. If matching OpenFlow rule is not present, untagged traffic is processed according to the default OpenFlow rule (drop or send to controller).

```
device(config-vlan-300)#no untagged ethernet 2/1
```

Case 4: When a port is removed as untagged from protected VLAN.

Configuration is accepted and untagged traffic on port 2/1 is forwarded as per OpenFlow rule, if matching rule is present. If matching OpenFlow rule is not present, untagged traffic is processed according to the default OpenFlow rule (drop or send to controller).

```
device(config-vlan-400)#no untagged ethernet 2/1
```

Case 5: An untagged configured unprotected VLAN is configured as protected VLAN on a port.

Configuration is accepted and untagged traffic on port 2/1 is forwarded as per the route table.

```
device(config-if-e10000-2/1)#openflow enable layer2 hybrid-mode
device(config-if-e10000-2/1)#vlan 300
device(config-vlan-300)#untagged ethernet 2/1
device(config-if-e10000-2/1)#openflow protected-vlans 300
```

Case 6: An untagged protected VLAN is removed from the port making it untagged configured unprotected VLAN.

Configuration is accepted and untagged traffic on the port 2/1, for example, is forwarded as per the OpenFlow rule if matching rule is present. If matching OpenFlow rule is not present, untagged traffic on port 2/1 is routed as per routing table. If route is not present, untagged traffic is processed according to the default OpenFlow rule (drop or send to controller).

```
device(config-if-e10000-2/1)#openflow enable layer2 hybrid-mode
device(config-if-e10000-2/1)#openflow protected-vlans 400
device(config-if-e10000-2/1)#vlan 400
device(config-vlan-400)#untagged ethernet 2/1
device(config-if-e10000-2/1)#no openflow protected-vlans 400
```

Case 7: Configured untagged unprotected VLAN is deleted.

Configuration is accepted and untagged traffic on port 2/1 is forwarded as per OpenFlow rule if matching rule is present. If matching OpenFlow rule is not present, untagged traffic is processed according to the default OpenFlow rule (drop or send to controller).

```
device(config)#no vlan 300
```

Case 8: Untagged protected VLAN is deleted.

Configuration is accepted and untagged traffic on port 2/1 is forwarded as per OpenFlow rule, if matching rule is present. If matching OpenFlow rule is not present, untagged traffic is processed according to the default OpenFlow rule (drop or send to controller).

```
device(config)#no vlan 300
```

Modifying flows

If an OpenFlow hybrid port is already a member of untagged non-default VLAN, flows with matched VLAN as untagged non-default VLAN is supported. But, these flows continue to be supported after changing the untagged VLAN.

- Flow with match VLAN as 0.
- A flow with Pop VLAN and same OpenFlow hybrid port in the action.

Untagged traffic and flow behavior

The behavior for untagged traffic matching untagged flow is as described below. For example, when VLAN 100 is port default or untagged VLAN of ingress port 2/1.

If both Input port (2/1) and output port (2/2) belong to the same untagged VLAN, the egress packet goes out as untagged.

```
ovs-ofctl add-flow tcp:10.37.66.11 --protocols=OpenFlow13 "in_port=49 dl_vlan=100 dl_type=0x800
actions=output:50
```

When both input port (2/1, VLAN 100) and output port (2/2, VLAN 200) belong to different untagged VLANs, the egress packet goes as tagged with tag value untagged VLAN on ingress port VLAN 100. If you want to send out the packet as untagged you may push the VLAN matching the untagged VLAN of egress port (VLAN 200).

```
ovs-ofctl add-flow tcp:10.37.66.11 --protocols=OpenFlow13 "in_port=49 dl_vlan=100 dl_type=0x800
actions=push_vlan:0x8100 set_field:200->vlan_vid output:50
```

When you push other VLAN (VLAN 300) in the above scenario, egress packet is tagged in the same VLAN.

```
ovs-ofctl add-flow tcp:10.37.66.11 --protocols=OpenFlow13 "in_port=49 dl_vlan=100 dl_type=0x800
actions=push_vlan:0x8100 set_field:300->vlan_vid output:50
```

Assumptions and limitations

1. At any time only one of the VLANs (protected VLAN or configured unprotected) can be made as an untagged member of an OpenFlow hybrid port.
2. Modifying flows is supported only on OpenFlow hybrid port.
3. When protected VLAN or configured unprotected VLAN is configured as untagged VLAN, flow matching untagged VLAN can be accepted.
4. Normal action for untagged traffic is not supported.

Idle and hard timeout support for OpenFlow

Each flow entry may have an idle timeout and or a hard timeout associated with it.

The idle timeout and a hard timeout control the removal of a flow entry from the OpenFlow table. If either value is non-zero, the switch must note the flow entry's arrival time, as it may need to evict the entry later. A non-zero idle_timeout entry field causes the flow entry to be removed after the given number of seconds, if no packet has been matched by the flow. A non-zero hard_timeout field causes the flow entry to be removed after the given number of seconds, regardless of how many packets it has matched.

The controller may actively remove flow entries from flow tables by sending delete flow table modification messages (OFPFC_DELETE or OFPFC_DELETE_STRICT). For modification requests (OFPFC_MODIFY or OFPFC_MODIFY_STRICT), if a matching entry exists in the table, the instructions field of this entry is updated with the value from the request, whereas its cookie, idle_timeout, hard_timeout, flags, counters, and duration fields are left unchanged.

When a flow entry is removed, either by the controller or the flow expiry mechanism, the switch must check the flow entry's OFPFF_SEND_FLOW_REM flag. If this flag is set, the switch must send a flow removed message to the controller. Each flow removed message contains a complete description of the flow entry, the reason for removal (expiry or delete), the flow entry duration at the time of removal, and the flow statistics at the time of removal.

In the case that a switch loses contact with all controllers, as a result of echo request timeouts, TLS session timeouts, or other events, the switch immediately enters either fail secure mode or fail standalone mode, depending upon the switch implementation and configuration.

In fail secure mode, the only change to switch behavior is that packets and messages destined to the controllers are dropped.

NOTE
Flow entries continue to expire according to their timeouts in fail secure mode in case of connection interruption.

In fail standalone mode, the switch processes all packets using the OFPP_NORMAL reserved port. In other words, the switch acts as a legacy Ethernet switch or router. The fail standalone mode is usually available only on hybrid switches. The existing flow entries remain, upon reconnecting to a controller. The controller then has the option of manipulating all flow entries, including deleting them.

The first time a switch starts up, it operates in either fail secure mode or fail standalone mode, until it successfully connects to a controller.

Port-based flow

The following shows the configuration of the flow.

```
dpctl tcp:10.37.66.24 flow-mod cmd=add,table=0,idle=5,hard=300,prio=222

in_port=5,eth_dst=00:00:00:00:bb:b1 vlan_vid=444 vlan_pcp=4 eth_type=0x800 apply:output:5
```

- When incoming traffic does not hit the flow, the idle time starts to increment. A hit causes this time to reset.
- The flow duration time keeps incrementing independent of the traffic hit.
- Once the limit is reached for either the idle time or the flow duration, the flow is deleted and a Syslog is generated indicating the reason for deletion (idle timeout, hard timeout).

Generic flow

The following shows the configuration of the flow.

```
dpctl tcp:10.37.66.24 flow-mod cmd=add,table=0,idle=5,hard=300,prio=222


eth_dst=00:00:00:00:bb:b1 vlan_vid=444 vlan_pcp=4 eth_type=0x800 apply:output:5
```

- Generic flows may be installed on multiple ports, depending on the OpenFlow-enabled ports. Even if the traffic hits only on one of those in_ports, flow statistics increments. Therefore, the idle time does not increment from its zero value.
- When incoming traffic does not hit any of the flows, the idle time starts to increment. A hit causes this time to reset.
- The flow duration time keeps incrementing independent of the traffic hit.
- Once the limit is reached for either the idle time or the flow duration, the flow is deleted and a Syslog is generated indicating the reason for deletion (idle timeout, hard timeout).

Limitations

1. The completion of the deletion of a flow in a system lags the timeout by a small amount, typically not to exceed a maximum of two minutes. The lag depends on the number of flows present in the system at that time.
2. The CER 2000 Series and CES 2000 Series support a maximum of 1,981 flows with idle timeout and hard timeout supported. It does not support generic flows with timeout fields.

   NOTE
   This number is shared between other applications and the OpenFlow flows without timeout set.

3. LP statistics and MP statistics may differ. Power-cycling or other reset events on an LP resets the LP statistics without impacting the information on the MP.
4. MP statistics is advertised, when requested by a controller.
5. The minimum timeout value for port-based flows is 15 seconds.
6. The minimum timeout value for generic flow is 30 seconds.

To show the timeout flows , enter the following command:

```
device(config)#show openflow flows timeout
Total Number of Active timeout flows                    :     1

Total Number of Active Idle(only) timeout flows                :     0
Total Number of Active Hard(only) timeout flows                :     0
Total Number of Active Idle + Hard(both) timeout flows         :     1

Total Number of Available Idle and/or Hard timeout flows    :  1980[METRO SPECIFIC]
```

To show the timeout flows in detail , enter the following command:

```
device(config)#show openflow flows timeout detail
Total Number of Active timeout flows                    :     1

Total Number of Active Idle timeout flows               :     0
Total Number of Active Hard timeout flows               :     0
Total Number of Active Idle + Hard timeout flows        :     1

Total Number of Available Idle and/or Hard timeout flows :  1980[METRO SPECIFIC]

Flow ID: 1 Priority: 222 Status: Active
        Rule:
```

```
   In Port:      generic
   In Vlan:      Tagged[444]
   Vlan PCP:     4
   Destination Mac:      0000.0000.bbb1
   Destination Mac Mask:         ffff.ffff.ffff
   Ether type:   0x00000800

Timing Info:
   Idle Timeout                          :   500 secs
   Hard Timeout                          :  3000 secs
   Time Elasped(Since Flow Added)        :   417 secs
   Time Elasped(Since Last Packet Hit)   :   417 secs

Instructions: Apply-Actions
        Action: FORWARD
                Out Port:  e1/5
Statistics:
   Total Bytes: 0
```

Layer 2 support for OpenFlow hybrid ports

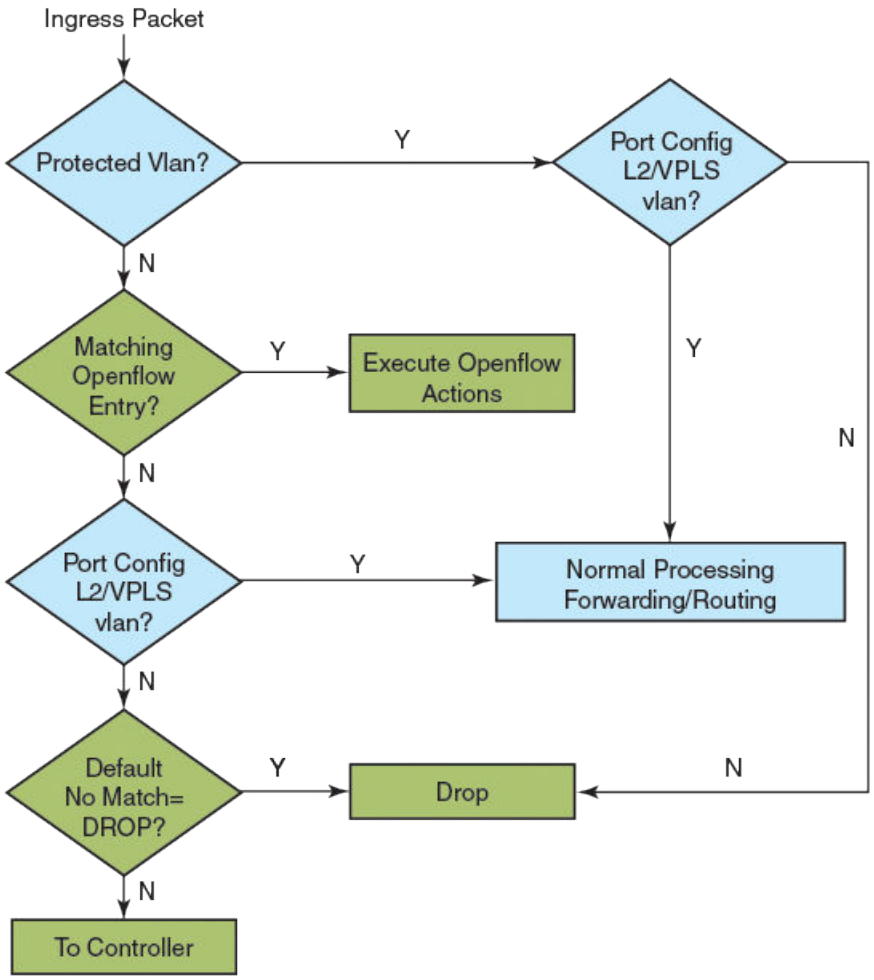
The XMR Series and MLX Series support Layer 2 protocols on the OpenFlow hybrid ports.

The following features are supported on the OpenFlow protected VLANs and configured unprotected VLANs for Layer 2.

- Layer 2 switching and MAC learning
- STP, SSTP, MSTP, RSTP, ERP
- LLDP, FDP, CDP
- LACP

Layer 3 routing on VE interface and VPLS switching or routing are supported with the Layer 2 support on the OpenFlow hybrid ports. The following diagram shows the flow for an ingress packet processed to reach the controller.

FIGURE 9 Packet flow diagram for Layer 2 support



Layer 2 switching and MAC learning

The SA MAC learning happens on the protected VLANs & configured unprotected VLANs. The unconfigured VLAN traffic is dropped or sent to the controller based on the default rule.

On untagged VLAN, untagged traffic is flooded and SA MAC is learnt. The tagged traffic matching untagged VLAN is dropped. If the untagged VLAN is made a protected VLAN, flow rules do not apply on untagged traffic, and the flow follows Layer 2 protocols. When the VLAN is configured a unprotected, then in the presence of flow, if the untagged traffic hits a matching flow rule, and the untagged traffic misses, it follows Layer 2 protocols.

When the tagged VLAN is protected VLAN, flow rules do not apply on matching tagged traffic, and it follows Layer 2 protocols. If it is configured unprotected VLAN, the flow matching tagged traffic hits flow rule in the presence of flow. In the absence of flow, it follows Layer 2 protocols.

This is an example to validate VLAN as tagged or untagged member on the non-hybrid OpenFlow ports.

```
device#show openflow interface

Total number of Openflow interfaces: 2

Port  Link     Speed Tag MAC            OF-portid  Name  Mode
2/1   Up       10G   Yes 0024.3880.8f30 49               Hybrid-Layer2
2/2   Up       10G   Yes 0024.3880.8f31 50               Layer2
device#
device#configure
device(config)#vlan 56
device(config-vlan-56)#tag ethernet 2/2
Error: Non Hybrid Openflow port 2/2 cannot be configured as tagged in Vlan 56
```

STP, SSTP, RSTP, MSTP, ERP

STP

The STP can be enabled globally on a per VLAN or per port basis. The STP runs on all the configured VLANs when enabled globally. When STP is enabled on a VLAN, all the ports (non OpenFlow or OpenFlow hybrid ports) become part of the STP instance. STP can be enabled on OpenFlow hybrid ports. The STP Bridge Protocol Data Units (BPDUs) are tagged packets, which contain VLAN ID on which the STP instance is running.

When you add a flow to match control packets, then it may affect the convergence of Layer 2. For proper protocol convergence, the VLANs running STP on OpenFlow hybrid ports should be protected VLANs.

Here is an example on enabling STP globally. A message is shown if OpenFlow non-hybrid ports are present.

```
device#show openflow interface
Total number of Openflow interfaces: 3
Port  Link     Speed Tag MAC            OF-portid   Name Mode
2/1   Up       10G   Yes 0024.3880.8f30 49               Hybrid-Layer2
2/2   Up       10G   Yes 0024.3880.8f31 50               Layer2
2/6   Disabled None  Yes 0024.3880.8f35 54               Layer2

device#configure
device(config)#spanning-tree
Openflow(Non Hybrid) port 2/2 has been excluded from STP.
Openflow(Non Hybrid) port 2/6 has been excluded from STP.
```

SSTP

You can configure the device to run a single spanning tree across all of the ports and VLANs. When SSTP is enabled, all the ports that are in port-based VLANs with STP-enabled become members of a single spanning tree domain. For proper SSTP protocol convergence, the untagged VLANs on OpenFlow hybrid ports should be protected VLANs. On enabling SSTP, a message is shown, if the OpenFlow non-hybrid ports are present.

```
device#show openflow interface
Total number of Openflow interfaces: 3
Port  Link     Speed Tag MAC            OF-portid   Name Mode
2/1   Up       10G   Yes 0024.3880.8f30 49               Hybrid-Layer2
2/2   Up       10G   Yes 0024.3880.8f31 50               Layer2
2/6   Disabled None  Yes 0024.3880.8f35 54               Layer2

device(config)#spanning-tree single
Openflow(Non Hybrid) port 2/2 has been excluded from SSTP.
Openflow(Non Hybrid) port 2/6 has been excluded from SSTP.
```

The OpenFlow non-hybrid port configuration is rejected if SSTP is running.

```
device(config)#spanning-tree single
Openflow(Non Hybrid) port 2/2 has been excluded from SSTP.

device(config)#
device(config)#int e 2/6
device(config-if-e10000-2/6)#open enable layer3 hybrid
device(config-if-e10000-2/6)#interface e 2/7
device(config-if-e10000-2/7)#open enable layer3
Error - Port 2/7 cannot be added as an openflow(Non Hybrid)port,
 due to SSTP configuration.
```

MSTP

MSTP protocol is supported only on the OpenFlow hybrid ports. For proper MSTP protocol convergence, the untagged VLANs on OpenFlow hybrid ports, should be protected VLANs.

RSTP

RSTP protocol is supported only on the OpenFlow hybrid ports. For proper protocol convergence, the VLANs running RSTP on OpenFlow hybrid ports should be protected VLANs.

ERP

When the VLANs are not protected, and you have a matching OpenFlow rule present, the BPDU is not processed. It takes the OpenFlow path and the protocol convergence is affected.

Case1 : ERP interfaces are configured on default VLAN.

```
erp 1
  left-interface vlan 1 ethernet 1/6
  right-interface vlan 1 ethernet 1/5
  raps-default-mac
  enable

device(config-if-e1000-1/6)#open enable
Error - ERP is enabled on vlan 1
```

Case 2 : ERP interfaces are configured on non-default VLAN.

```
erp 1
  left-interface vlan 10 ethernet 2/1
  right-interface vlan 10 ethernet 2/2
  raps-default-mac
  enable

device(config-if-e10000-2/1)#open enable
Error: Port 2/1 is tagged in some vlan, can't be enabled Openflow(Non-Hybrid) on it
```

LLDP, FDP, CDP

These protocols are supported on the OpenFlow hybrid ports. In case the untagged VLAN are protected, protocol convergence is achieved. When untagged VLAN is not protected and a matching OpenFlow rule is present, the PDUs do not act processed. It takes the OpenFlow path and the protocol convergence is affected.

The following example shows that LLDP configurations are rejected on non-hybrid ports.

```
device#show openflow interface
Total number of Openflow interfaces: 3

Port  Link     Speed Tag MAC            OF-portid   Name Mode
2/1   Up       10G   Yes 0024.3880.8f30 49               Hybrid-Layer2
2/2   Up       10G   Yes 0024.3880.8f31 50               Layer2
2/8   Up       10G   Yes 0024.3880.8f37 56               Hybrid-Layer2

device#configure
device(config)#lldp enable ports ethernet 2/1 to 2/8
Error: Port 2/2 is excluded from LLDP, as openflow(Non Hybrid) is enabled on it
```

LACP

Dynamic LAG is supported on the OpenFlow hybrid ports. You must complete the configuration in the following sequence to avoid rejection.

1. Deploy LAG
2. Configure OpenFlow (and protected VLANs) on primary LAG port.
3. Controller can configure flows on individual LAG ports.
4. Unconfigure the OpenFlow (and protected VLANs) on primary LAG port.
5. Un-deploy LAG.

You must do the following to configure the OpenFlow.

```
device#show openflow interface
Total number of Openflow interfaces: 4

Port  Link     Speed Tag MAC            OF-portid Name Mode
2/1   Up       10G   Yes 0024.3880.8f30 49             Hybrid-Layer2
2/2   Up       10G   Yes 0024.3880.8f31 50             Layer2
2/3   Up       10G   No  0024.3880.8f32 51             Hybrid-Layer2
2/6   Disabled None  No  0024.3880.8f35 54             Hybrid-Layer2

device#configure
device(config)#lag lag2 dynamic id 3
device(config-lag-lag2)#ports ethernet 2/1
Error: Openflow configuration exist on port 2/1
```

Un-deployed LAG ports cannot be added as OpenFlow hybrid ports.

```
device(config-lag-lag2)#ports ethernet 2/4
device(config-lag-lag2)#
device(config-lag-lag2)#interface ethernet 2/4
device(config-if-e10000-2/4)#openflow enable layer2 hybrid
Error - Port 2/4 is either undeployed LAG port or not a LAG primary port
```

OpenFlow (non-hybrid) is not enabled on deployed LAG ports.

```
device#(config-lag-lag3)#show openflow interface

Total number of Openflow interfaces: 4

Port  Link     Speed Tag MAC            OF-portid   Name Mode
2/1   Up       10G   Yes 0024.3880.8f30 49               Hybrid-Layer2
2/2   Up       10G   Yes 0024.3880.8f31 50               Layer2
2/3   Up       10G   No  0024.3880.8f32 51               Hybrid-Layer2
2/6   Disabled None  No  0024.3880.8f35 54               Hybrid-Layer2
device(config-lag-lag3)#ports ethernet 2/7
device(config-lag-lag3)#primary-port 2/7
device(config-lag-lag3)#deploy
device(config-lag-lag3)#interface ethernet 2/7
```

```
device(config-if-e10000-2/7)#openflow enable
Error - Port 2/7 is part of LAG configuration, so Non-Hybrid Openflow can't be enabled
```

MCT support for OpenFlow hybrid ports

The XMR Series and MLX Series support MCT on OpenFlow hybrid ports.

Ports on MCT peer switches are classified into one of the following three: CCEP ports, CEP ports or ICL ports.

The following are supported on OpenFlow hybrid ports.

- Layer 2 MCT
- L2VPN MCT (VPLS MCT)

CCEP ports

CCEP ports are generally members of the LAG interface to the MCT client. These ports can be enabled as OpenFlow hybrid ports without any exceptions. All the VLANs configured on this port, becomes protected or unprotected VLANs. In the absence of an OpenFlow flow, CCEP ports functionality work properly. If you make the VLANs protected, CCEP works on those VLANs as well. If these VLANs are configured unprotected, then matching flow rule takes effect.

CEP ports

CEP ports are general ports on MCT peer switches. They are neither Cluster Client Edge Port (CCEP) nor an ICL port. These ports can be enabled as OpenFlow hybrid ports without any exceptions. All the VLANs configured on this port, become protected or unprotected VLANs. If you make the VLANs protected, CEP functionality works on those VLANs as well. If these VLANs are configured unprotected, then matching flow rule takes effect.

ICL ports

A single port or multi-port interface between the two MCT peer switches, ICL ports should not be untagged members of any VLAN. On ICL ports, non-MCT VLANs can coexist with MCT VLANs.

Session VLANs operate on ICL links. CCP protocol runs over this VLAN.

1. It is recommended that the Session VLAN be a protected VLAN, to ensure control packets are not affected by OpenFlow rule.
2. All other VLANs can be protected or unprotected.
3. Matching OpenFlow rule takes precedence in case of unprotected VLANs, whereas protected VLANs is protected from OpenFlow traffic.

VRF support for OpenFlow hybrid ports

The XMR Series and MLX Series support VRF on OpenFlow hybrid ports. VRF is supported on virtual interfaces (VE), VE over Layer 2 VLANs as well as over VPLS VEs.

In absence of flows, VRF works normally on OpenFlow hybrid ports. If you create a VLAN which is used for a VRF instance, as a protected VLAN, then OpenFlow flows bypass and VRF functionality is effective. If the VLAN which is used for a VRF instance is configured unprotected, and a matching OpenFlow rule is present (wild-carded VLAN rule or configured unprotected VLAN rule), the packet takes the path defined by the OpenFlow rule.

Limitations

If a physical port has been used in a VRF instance, non-hybrid OpenFlow port cannot be configured on that port, and it generates an error message. The example below shows the error message.

```
interface ethernet 1/1
enable
vrf forwarding green
ip ospf area 1
ip address 100.1.1.1/24

device(config-if-e1000-1/1)#open enable
Error: Cannot enable on port: 1/1 with Layer 3 config.
```

NOTE

This feature is not supported on 10x24GbE interface module.

ACL and PBR support for OpenFlow hybrid ports

Access Control List (ACL) and Policy Based Routing (PBR) are supported on OpenFlow hybrid interfaces along with OpenFlow flows.

The configured protected VLAN traffic follows the OpenFlow rules as well as ACL and PBR rules. Unconfigured protected VLAN traffic does not follow the ACLs or PBR rules and the traffic is dropped.

The traffic with configured unprotected VLAN follows OpenFlow rules, if there are no rules to forward the unprotected VLAN traffic then it follows ACL and PBR rules. The traffic on an OpenFlow hybrid port with unconfigured unprotected VLAN is dropped or sent to the OpenFlow controller.

The traffic with configured unprotected VLAN follows the normal action flow and OpenFlow rules, if there are no rules to forward the unprotected VLAN traffic then it follows ACL and PBR rules.

OpenFlow forwarding behavior

The ingress traffic on an OpenFlow interface is modified as per OpenFlow rule and routing entry, if present. For traffic matching OpenFlow rule, traffic is redirected as per the rule, however if a matching route entry is present for the same traffic, the IP route action is picked up from the route entry and the packet is modified accordingly even if the OpenFlow rule does not specify such action. The action in OpenFlow rule overrides the action from route entry action, if present.

For the following examples, the IP packet is icoming in on VLAN 10 with a Destination IP of 172.25.0.3 on physical port 1/1. The incoming traffic matches both the IP route entry and the OpenFlow rule entry.

Case 1: OpenFlow rule -> Match (VLAN 10, Destination IP: 172.25.0.3) -> Action (Outport:12/1)

Due to matching route entry being present along with matching OpenFlow rule, outgoing packet VLAN gets modified to VLAN 20 and Destination MAC is modified to 0024.38ef.1c50 as per route entry while being sent out on port 12/1 following OpenFlow rule.

Case 2: OpenFlow rule -> Match (VLAN 10, Destination IP: 172.25.0.3) -> Action (Mod VLAN: 30, mod_dst_mac: 00:00:00:01:02:03, Outport:12/1)

OpenFlow rule specifies VLAN and Destination MAC in action. Outgoing packet VLAN gets modified to VLAN 30 and Destination MAC is modified to 00:00:00:01:02:03 as per OpenFlow rule while being sent out on port 12/1.

Case 3: OpenFlow rule -> Match (VLAN 10, Destination IP: 172.25.0.3) -> Action (Outport:12/1)

In the absence of matching IP route entry present, traffic behaves as per OpenFlow rule. Outgoing packet VLAN and Destination MAC are not modified and traffic is redirected to port 12/1.

Limitations

- The traffic on a Layer 2 OpenFlow hybrid port follows only Layer 2 PBR and ACL rules.
- The traffic on a Layer 3 OpenFlow hybrid port follows only Layer 3 PBR and ACL rules.
- PBR and ACL are not supported on L23 OpenFlow hybrid port.
- Layer 2 PBR and ACLs are not supported on Layer 3 OpenFlow hybrid port.
- Layer 3 PBR and ACLs are not supported on Layer 2 OpenFlow hybrid port.
- Layer 2 ACL deny logging is not supported.
- Traffic hitting an OpenFlow rule, gets modified as per the route entry present, but with OpenFlow rules with high priority.
- When OpenFlow and allow-all-vlan PBR are configured on the same port, OpenFlow properties have higher priority.

Configuring ACL and PBR for OpenFlow interfaces

To enable PBR and ACLs for Openflow interface, use the following command.

```
openflow enable ofv130 [acl-pbr]
```

The following example shows the ACL and PBR interfaces for **show openflow** command.

```
device(config)#show openflow

Administrative Status:       Enabled
SSL Status:                  Enabled
Source-Interface:            Not Configured
Source-Interface Status:     NA
Controller Type:             ofv130
ACL/PBR Status:                  Enabled
HELLO Reply:                 enabled
Number of Controllers:       0

Match Capability:

L2 : Port, Source MAC, Destination MAC, Ether type, Vlan, Vlan PCP
     MPLS Label, MPLS BoS

L3 : Port, Vlan, Vlan PCP, Ethertype(IP,ARP,LLDP), Source IP, Destination IP,
     IP Protocol, IP TOS, IP Src Port, IP Dst Port
     Ethertype(IPV6), Source IPv6, Destination IPv6, ICMPv6 Type, ICMPv6 code, ICMPv4 Type, ICMPv4 code,
MPLS Label, MPLS BoS

L23: All

Normal Openflow Enabled Ports:
Openflow Hybrid Physical Interfaces:
e4/1
Protected VLANs   : None
Unprotected VLANs :     1,
e4/3
Protected VLANs   : None
Unprotected VLANs :     1,
Openflow Hybrid Logical Interfaces:

IPSEC(name/of-portid):
```

Local port mirroring

Local port mirroring allows you to replicate hybrid VLAN traffic on OpenFlow hybrid port to another OpenFlow port (mirror port) without affecting normal forwarding of traffic.

Routed or switched traffic on configured unprotected VLANs can be replicated to another port using OpenFlow rules. In short, configured unprotected VLAN traffic matching Normal action flow along with mirror port is supported.

NOTE

To support local port mirroring, you can provide an output port along with Normal action.

A sample flow is given below, where **output:50** is the mirror port for the flow.

```
ovs-ofctl add-flow tcp:10.25.106.60 --protocols=OpenFlow13 "in_port=52 dl_type=0x800 dl_vlan=200
priority=34000 actions=normal output:50"
```

Considerations for configuring a mirror port

Port configured as **acl-mirror-port** has to be an OpenFlow port since the same port needs to be specified while configuring the flow. This configuration can be supported on a LAG port as well but **acl-mirror-port** must be configured at the LAG level for individual LAG member ports.

While adding the flow with Normal action and a mirror port, the mirror port should match with the **acl-mirror-port** configured on the ingress port. Otherwise the flow is rejected with an error message. While removing the **acl-mirror-port** configuration on the ingress port, an OpenFlow rule with Normal action and mirror-port should not be present on that ingress port. Otherwise the configuration is rejected with an error message.

Assumptions

1. Monitor port (ingress OpenFlow port) can be a physical port or a LAG port.
2. Mirror port can be from same or different PPCR.
3. Supported only on OpenFlow hybrid ports (configured unprotected VLAN).

Limitations

1. OpenFlow flows belonging to a particular PPCR (having ingress port from the same PPCR) have a common mirror port.
2. Limitation of Normal action flow is applicable except that an output port can be specified along with action Normal.
3. Mirror port is a physical port thus remote traffic mirroring cannot be supported.
4. Port-based mirroring and flow-based mirroring are mutually exclusive.
5. Mirror port in OpenFlow rule does not accept any other action such as modify VLAN, MAC, etc.
6. Destination mirror port is a physical OpenFlow port.
7. Generic flow with Normal action and mirror port is not supported.

Displaying the OpenFlow flows

You can display the OpenFlow flows that are configured on the devices with IPv6 fields. The **show openflow flows** command shows all the flows configured in the system flow table. If you specify the interface, all the flows configured in the system for that interface are displayed.

Below is an example to show the IPv6 fields for both in MP and LP. Enter the following command:

```
device(config)# show openflow flows
Flow ID: 8 Priority: 32768 Status: Active
        Rule:
           In Port:      generic
           Ether type:   0x000086dd
           IP Protocol:          58
           Source IPv6:          2001:db8:85a3:42:0:8a2e:370:7334/128
           Destination IPv6:     2001:2:6c::430/48
           ICMPV6 Type:          1
           ICMPV6 Code:          3
        Instructions: Apply-Actions
                Action: FORWARD
                        Out Port:  e1/15
        Statistics:
           Total Bytes: 0
```

show openflow flows have ICMPv4 fields in match for both in MP and LP as following.

```
device(config)# show openflow flows
Flow ID: 6 Priority: 32768 Status: Active
        Rule:
           In Port:      generic
           Ether type:   0x00000800
           Source IP:            51.1.1.1       Subnet IP:       255.255.255.255
           Destination IP:       40.1.1.1       Subnet IP:       255.255.255.0
           IP Protocol:          1
           ICMPV4 Type:          3
           ICMPV4 Code:          2
        Instructions: Apply-Actions
                Action: FORWARD
                        Out Port:  e1/16
        Statistics:
           Total Bytes: 0
```

The **show openflow flows** command has the match capability as below.

Layer 2: Port, Source MAC address, Destination MAC address, Ether type, VLAN, VLAN PCP

Layer 3 : Port, VLAN, VLAN PCP, Ether type (IP, IPv6 ,ARP and LLDP), Source IP address, Destination IP address, IP Protocol, IP TOS , IP Source Port, IP Destination Port, Source IPv6 address, Destination IPv6 address, ICMPv6 Type, ICMPv6 code ICMPv4 Type, ICMPv4 code

Layer23: All

> **NOTE**
> IP Protocol represents both IPv4 and IPv6 protocol numbers depending on the matching rules for IPv4 or IPv6 fields.

> **NOTE**
> Matching IP TOS supports both IPv4 DSCP and IPv6 traffic class.

Asynchronous configuration

Asynchronous messages may need to be sent to multiple controllers. An asynchronous message is duplicated for each eligible OpenFlow channel, and each message is sent when the respective controller connection allows it.

A controller can also control which types of switch asynchronous messages are sent over its OpenFlow channel. This is done using an asynchronous configuration message that has the filter setting for all the messages.

Different controllers can receive different notifications. A controller in the Master role can selectively disable notifications, and a controller in the Slave role can enable notifications it wants to monitor.

Each controller configuration block for active connection maintains its own asynchronous configuration setting for every role. The default initial configuration is shown in the following table.

TABLE 27 Action for asynchronous configuration

Messages	Bit field	Master or Equal role	Slave role
Packet-in reasons	No_match	Enable	Disable
	Action	Enable	Disable
	Invalid_TTL	Enable	Disable
Port status reasons	Add	Enable	Enable
	Delete	Enable	Enable
	Modify	Enable	Enable
Flow removed reasons	Idle_timeout	Enable	Disable
	Hard_timeout	Enable	Disable
	Delete	Enable	Disable
	Group_delete	Enable	Disable

NOTE

The asynchronous messages Action and Invalid_TTL are not supported by these devices. Controllers can set these bits in the filter setting and the device can accept the bits, but the messages are not sent out by the device.

Normal action

Normal action represents the traditional routing with traffic on configured protected and unprotected VLAN with OpenFlow hybrid port. It can be used only as an output port and it processes the packet using the normal flows.

OpenFlow flows can be configured with action as Normal matching configured protected and unprotected VLAN on an OpenFlow hybrid port. Traffic matching the configured protected and unprotected VLAN on that hybrid port is subjected to forwarding or routing as per the forwarding table. QoS parameter such as meters, however is taken from the matching flow entry and applied to the traffic.

When the device loses contact with all controllers, the device immediately enters into the fail secure mode or fail standalone mode. In fail secure mode, the packets and messages for the controller are dropped for the device. In fail standalone mode, the device processes all packets using the OFPP_NORMAL reserved port. Devices support only the fail secure mode.

Considerations

The following assumptions are taken into consideration for Normal action flows.

1. If route or forwarding entry is not present for configured protected and unprotected VLAN, the packet is dropped even though a matching Normal action flow exists.
2. Multiple Normal action flows can be configured for the same match VLAN with different QoS action. Traffic matches the highest priority flow among normal action flows.
3. Flow with different action for the same match VLAN as Normal action flow can also be configured.
4. When multiple flows are present, traffic hits the flow with the highest priority.

Limitations

These are the limitation for Normal action flows.

1. Though output port (physical port, drop or controller) in the action cannot be specified along with Normal action, mirror port can be specified.
2. Generic flow with Normal action is not supported. Only port based flows are supported.
3. Wild card VLAN match (port based or generic) flows with Normal action is not supported.
4. Normal action flow is supported only on OpenFlow hybrid port.
5. Only meter and mirror actions are supported for Normal action flows.

   NOTE
   For a flow with action other than Normal but matching Normal action flow in VLAN on an OpenFlow hybrid port, if that flow does not specify a Destination MAC address , or a VLAN or VLAN PCP in the action list, these fields still get updated if there is a matching IP route entry.

Updating show OpenFlow flows output

show openflow flows shows the Normal action in the output at MP.

```
device#show openflow flows
Total Number of data packets sent to controller:                    0
Total Number of data bytes sent to controller  :                    0

Total Number of Flows: 1
        Total Number of Port based Flows         : 1

        Total Number of L2 Generic Flows         : 0
        Total Number of L3 Generic Flows         : 0
        Total Number of L2+L3 Generic Flows      : 0
        Total Number of L23 Generic Flows        : 0

Total Number of Hardware entries for flows: 1
        Total Number of Hardware entries for Port flow: 1
        Total Number of Hardware entries for Generic flow: 0

Total Number of Openflow interfaces: 2
        Total Number of L2  interfaces: 0
        Total Number of L3  interfaces: 0
        Total Number of L23 interfaces: 2

Flow ID: 1 Priority: 34000 Status: Active
        Rule:
           In Port:      e3/1
           In Vlan:      Untagged
           Ether type:   0x00000800
        Instructions: Apply-Actions
                Action: FORWARD
                        Out Port: NORMAL
        Statistics:
           Total Pkts: 21943533
           Total Bytes: 1492160244
```

show openflow flows shows the normal action in the output at LP.

```
device#show openflow flows
Total Number of data packets sent to controller:                    0
Total Number of data bytes sent to controller  :                    0

Total Number of Flows: 1
Flow Id: 6, Priority: 32768, FD Id: 0, PW Id: 1
        Rule:
           In Port:      e3/1
           In Vlan:      Untagged
           Ether type:   0x00000800
        Action: FORWARD
                Out Port:       NORMAL
           FID: -N/A-, MVID: -N/A-
        Hardware Information:
        Port: 3/4   PPCR Id : 3, CAM Index: 0x000a50de (L4)  PRAM Index: 0x0007ff5c Packets: 0
        Statistics:
           Total Pkts: 0
           Total Bytes: 0
```

Configuring a flow with Normal action

When you add flows with Normal action, configure the device in the following order.

Add configured protected and unprotected VLAN on an OpenFlow hybrid port.

1. Enable OpenFlow globally.

```
device(config)# openflow enable ofv130
```

2. Configure system-max for OpenFlow entries.

```
device(config)# system-max openflow-flow-entries 65536
```

3. Configure system-max for protected and unprotected VLAN entries.

```
device(config)system-max openflow-pvlan-entries 200
device(config)system-max openflow-unprotectedvlan-entries 200
```

4. Configure NP system-max for OpenFlow entries for the required slot.

```
device (config)system-max np-openflow-flow-entries layer2or3 1000 layer23ipv4 2000 slot 2
```

5. Configure OpenFlow Controller.

```
device (config)openflow-flow-entries layer2or3 1000 layer23ipv4 2000 slot 2
```

6. Enable OpenFlow on the port.

```
device(config-if-e1000-3/1)openflow enable layer2 hybrid
```

7. Push Normal action flow with untagged VLAN.

```
ovs-ofctl add-flow tcp:10.24.5.135 --protocols=OpenFlow13 " in_port =97,dl_type=0x800,dl_vlan=0xffff
actions=normal"
```

Rate limiting and mirroring for Normal action on Openflow ports

The rate limiting works as tagged VLAN flows when VLAN ID is 0xffff, which makes it an untagged VLAN flow.

A sample flow is as shown below for untagged OpenFlow hybrid port for rate limiting feature.

```
/usr/local/bin/ovs-ofctl  --protocols=OpenFlow13 add-meter tcp:172.24.69.11
"meter=1,kbps,burst,band=type=drop,rate=3000,burst_size=300000"
/usr/local/bin/ovs-ofctl  --protocols=OpenFlow13 add-flow tcp:172.24.69.11 "priority=32000 in_port=15
dl_type=0x0800 dl_vlan=0xffff actions=meter:1,normal,1"
```

The following untagged OpenFlow Normal actions are supported starting NetIron 5.9.00a release.

- Untagged OpenFlow hybrid port with Normal meter.
- Untagged OpenFlow hybrid port with Normal mirror.
- Untagged OpenFlow hybrid port with Normal meter and mirror.

The tagged OpenFlow Normal meter and mirror actions are already supported in prior releases.

An example for Normal action with meter is shown as below:

```
ovs-ofctl add-flow tcp:10.24.5.135 --protocols=OpenFlow13 "in_port=97,dl_type=0x800,dl_vlan=0xffff
actions=meter:1,normal".
```

If such a flow exists on the port, untagged traffic received on hybrid port get this flow and is forwarded in a traditional way and replicates traffic out of the same port.

```
/usr/local/bin/ovs-ofctl  --protocols=OpenFlow13 add-flow tcp:172.24.69.11 "priority=32000 in_port=15
dl_type=0x0800 dl_vlan=0xffff actions=normal,1"
```

The port mirroring is applicable for monitoring real time traffic. An untagged match flow can be created using following OVS command for normal action with mirror port flow.

```
ovs-ofctl add-flow tcp:10.24.5.135 --protocols=OpenFlow13 "in_port=97,dl_type=0x800,dl_vlan=0xffff
actions=normal,output:97".
```

An example for Normal action with meter and mirror port flow is shown as below:

```
ovs-ofctl add-flow tcp:10.24.5.135 --protocols=OpenFlow13 "in_port=97,dl_type=0x800,dl_vlan=0xffff
actions=normal,output:97".
```

For rate limiting and mirroring, the mirrored traffic must also be rate-limited. If such a flow exists on the port, untagged traffic received on hybrid port is forwarded in a traditional way with rate limit specified by OpenFlow meter and replicates traffic out of the same port .

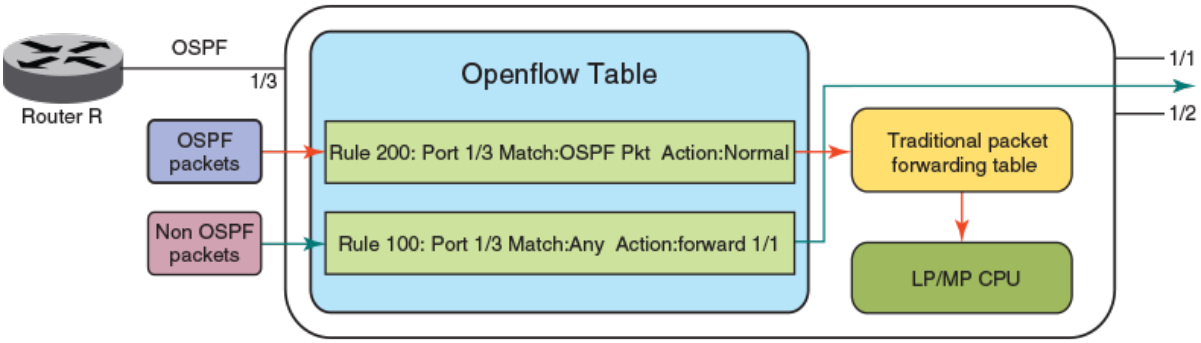
```
/usr/local/bin/ovs-ofctl  --protocols=OpenFlow13 add-meter tcp:172.24.69.11
"meter=1,kbps,burst,band=type=drop,rate=3000,burst_size=300000"
/usr/local/bin/ovs-ofctl  --protocols=OpenFlow13 add-flow tcp:172.24.69.11 "priority=32000 in_port=15
dl_type=0x0800  dl_vlan=0xffff actions=meter:1,normal,1"

Flow ID: 52 Priority: 32000 Status: Active
        Rule:
           In Port:      e1/9
           Ether type:   0x00000800
           Idle Timeout                          :     0 secs
           Hard Timeout                          :     0 secs
         Instructions: Apply-Actions
                Action: FORWARD
                        Out Port:  e1/1
                        Out Port: NORMAL
                Meter id: 1
```

The Normal action protects the control traffic in the hybrid port mode deployment following the OpenFlow rules.

Normal action for control traffic

By default, both control and user traffic are subjected to OpenFlow rules manipulation. To avoid the control protocol packets to be affected by the OpenFlow rules, the OpenFlow rules with Normal action can forward the control protocol packet through normal forwarding path and CPU.

In the following example, OSPF is enabled between device and router R. An OpenFlow rule 200 is created to match with the OSPF packets and it forwards using traditional packet forwarding table where the packets are further directed to CPU to either LP or MP. The subsequent OpenFlow rules at lower priority do not affect the OpenFlow packet.



OpenFlow rule 32001 is inserted to make sure OSPF packet is processed by system but not dropped by the subsequent OpenFlow rules.

```
ovs-ofctl add-flow  tcp:172.24.69.11 --protocols=OpenFlow13 "priority=32001 in_port=15 dl_type=0x0800
dl_vlan=0xffff nw_proto=89 actions=NORMAL"

Flow ID: 11 Priority: 32000 Status: Active
        Rule:
           In Port:      e1/15
           Ether type:   0x00000800
           Idle Timeout                          :     0 secs
           Hard Timeout                          :     0 secs
       Instructions:
```

```
                Action: DROP
            Statistics:
               Total Pkts: 20
               Total Bytes: 2224
            Timing Info:
               Time Elapsed(Since Flow Added)       :   125 secs
               Time Elapsed(Since Last Packet Hit)  :    26 secs

    Flow ID: 12 Priority: 32001 Status: Active
            Rule:
               In Port:      e1/15
               Ether type:   0x00000800
               IP Protocol:          89
               Idle Timeout                         :     0 secs
               Hard Timeout                         :     0 secs
             Instructions: Apply-Actions
                    Action: FORWARD
                            Out Port: NORMAL
            Statistics:
               Total Pkts: 18
               Total Bytes: 2216
            Timing Info:
               Time Elapsed(Since Flow Added)       :    19 secs
               Time Elapsed(Since Last Packet Hit)  :     0 secs
```

Supporting MPLS for OpenFlow

MPLS support is based on MPLS node type for services like IPv4 over MPLS, IPv4-L3VPN and L2VPN for OpenFlow v1.3.0.

To support MPLS services for different OpenFlow match fields and actions, these are the supported MPLS node types:

- Ingress MPLS node
- Transit MPLS node
- Egress MPLS node

TABLE 28 Supported match fields and actions for nodes

Node type	Match fields	Actions
Ingress MPLS node	Layer 2 or Layer 3 fields (Source MAC address, Destination MAC address, Ether type, VLAN, VLAN PCP, Source IP, Destination IP, IP Protocol, IP TOS, IP Source Port, IP Destination Port) and input port. **NOTE** It is based on OpenFlow interface type (Layer 2/Layer 3/L23) **NOTE** Flows matching Ether type as 0x800 can only support IPv4 over MPLS or IPv4-L3VPN.	• Push one label (Tunnel label) • Push two labels (Tunnel and service label) • Replace Destination MAC address • Set MPLS EXP • Push one VLAN (only for L2VPN) • Modify or Set outer VLAN (only for IPv4 over MPLS and IPv4-L3VPN) • Copy TTL out (only for IPv4 over MPLS and IPv4-L3VPN) **NOTE** When the action contains MPLS label match L2VPN service label, a new, Layer 2 header is inserted and the ingress packet is copied to payload at the egress. **NOTE** When destination MAC address is not specified in the action, the destination MAC address is set to 0.
Transit MPLS node	Match with one label (with MPLS Ether type 0x8847 or 0x8848) Match with Bottom of Stack (BoS) 0 or 1.	• Forward without any action • Pop one label • Modify or Set label • Push one label

TABLE 28 Supported match fields and actions for nodes (continued)

Node type	Match fields	Actions
		• Replace Destination MAC address • Set MPLS EXP • Send out of specific ports • Send out from multiple interfaces by swapping different label on each output port (MVID-based forwarding) • Send out from multiple interfaces by pushing different VLANs • Replace VLAN (Optional) • Send to controller • Drop • Decrement TTL (default action in forwarding plane even without TTL actions)
Egress MPLS node	Match with one label (with MPLS Ether type 0x8847 or 0x8848) Match with Bottom of Stack (BoS) 0 or 1.	• Pop one label • Send out of specific ports • Send out from multiple interfaces by pushing different VLANs • Modify or Set VLAN • Replace Destination MAC address • Copy TTL • Decrement TTL • Send to controller • Drop **NOTE** When the match field contains MPLS label match L2VPN service label, the new, outer Layer 2 header is removed and the payload becomes the packet at the egress.

NOTE
For detailed information on MPLS, please refer to *NetIron MPLS Configuration Guide.*

Data flow

When the first MPLS label is pushed, Ether type in Layer 2 header changes to MPLS Ether type. When the last MPLS label is popped, Ether type in Layer 2 header changes to inner header type (IPv4 Ether type for IP over MPLS and IPv4 VPN).

TABLE 29 Action taken at the node for data flow

MPLS nodes	Layer 2 header		VLANS	
	Source MAC	Destination MAC	Outer VLAN	Inner VLAN
Ingress	Modifies Source Address to Interface MAC	Modifies to Destination Address from next hop	Deletes incoming VLAN by default and takes outgoing VLAN from next hop	Deletes incoming VLAN by default and no inner VLAN for outgoing packet
Transit	Modifies Source Address to Interface MAC	Push: Modifies to Destination Address from next hop Swap/Pop/Others: Modifies to Destination Address from OpenFlow rule	Push: Deletes incoming VLAN by default and takes outgoing VLAN from next hop Swap/Pop/Others: Removes incoming VLAN and adds VLAN based on OpenFlow rule	Push: Deletes incoming VLAN by default and no inner VLAN for outgoing packet Swap/Pop/Others: Removes incoming VLAN and adds VLAN based on OpenFlow rule
Egress	Modifies Source Address to Interface MAC	Modifies to Destination Address from flow or don't modify Destination Address	Removes incoming VLAN and adds VLAN based on OpenFlow rule	Removes incoming VLAN and adds VLAN based on OpenFlow rule

IPv4overMPLS configuration for OpenFlow

FIGURE 10 Packet modification at each node for IPv4 over MPLS

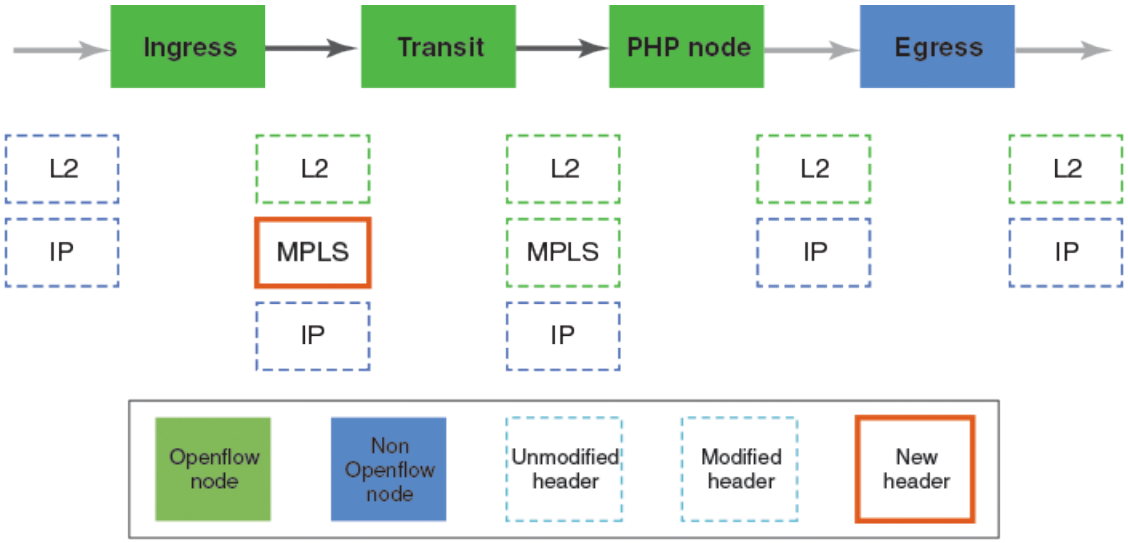


TABLE 30 Action taken at the node for IPv4 over MPLS

Ingress Node	Transit node	PHP node*
• Push over MPLS label • Modify Layer 2 header • Delete incoming VLANS and add up to one new VLAN • Flow – Match: Layer 2/Layer 3/VLAN/ Port fields – Action: Push one MPLS label, Set EXP and copy TTL out	• Modify MPLS header • Modify Layer 2 header • Delete incoming VLANS and add up to one new VLAN • Flow – Match: MPLS label – Action: Swap MPLS label	• Pop MPLS label • Modify Layer 2 header • Modify Ether type • Flow – Match: MPLS label – Action: Pop MPLS label, copy TTL in

NOTE

* : PHP node is Penultimate Hop Popping.

IPv4-L3VPN configuration for OpenFlow

FIGURE 11 Packet modification at each node for IPv4-L3VPN

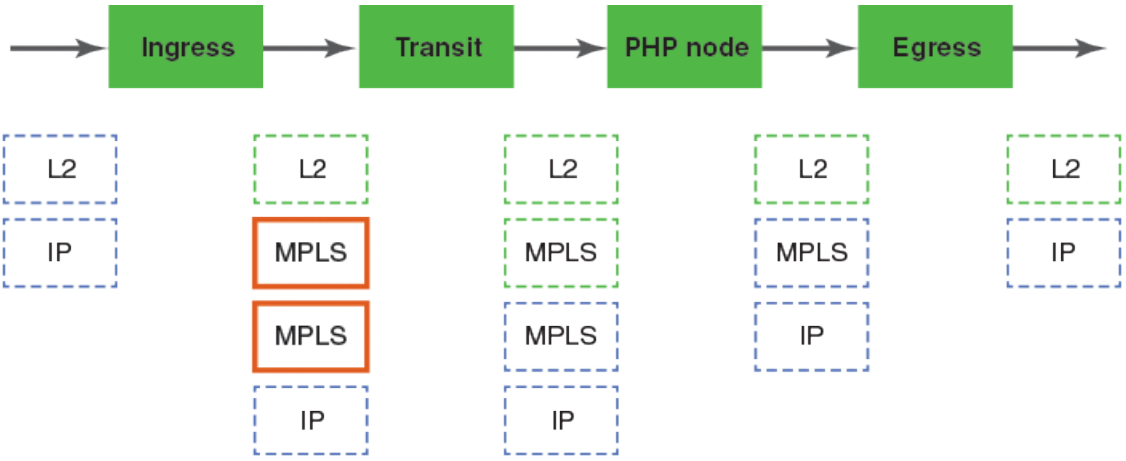


TABLE 31 Action taken at the node for IPv4-L3VPN

Ingress Node	Transit node	PHP node	Egress node
• Push two MPLS labels • Modify Layer 2 header • Delete incoming VLANS and add up to one new VLAN • Flow – Match: Layer 2/ Layer 3/VLAN/ Port fields – Action: Push two MPLS labels, Set EXP and copy TTL out	• Modify MPLS outer header • Modify Layer 2 header • Delete incoming VLANS and add up to one new VLAN • Flow – Match: MPLS label – Action: Swap MPLS label, Set EXP	• Pop outer MPLS label (Tunnel label) • Modify Layer 2 header • Modify Ether type • Flow – Match: MPLS label – Action: Pop MPLS label, copy TTL in	• Pop outer MPLS label (Tunnel label) • Modify Layer 2 header • Modify Ether type • Flow – Match: MPLS label – Action: Pop MPLS label, copy TTL in

L2VPN configuration for OpenFlow

FIGURE 12 Packet modification at each node for L2VPN

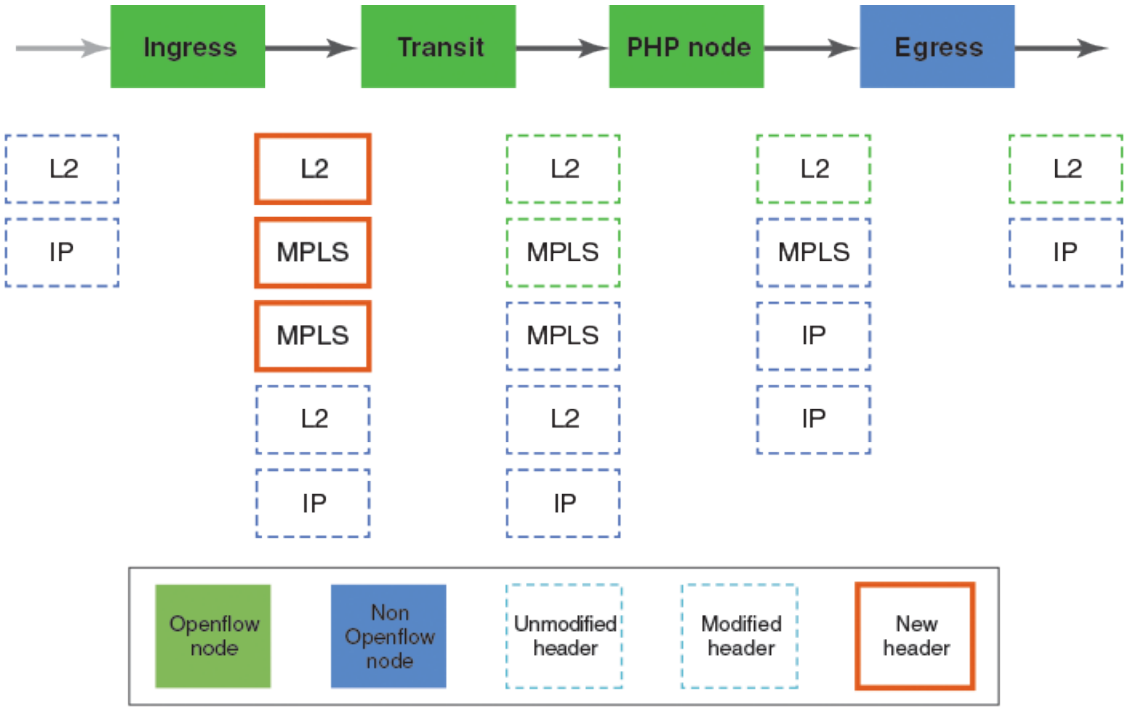


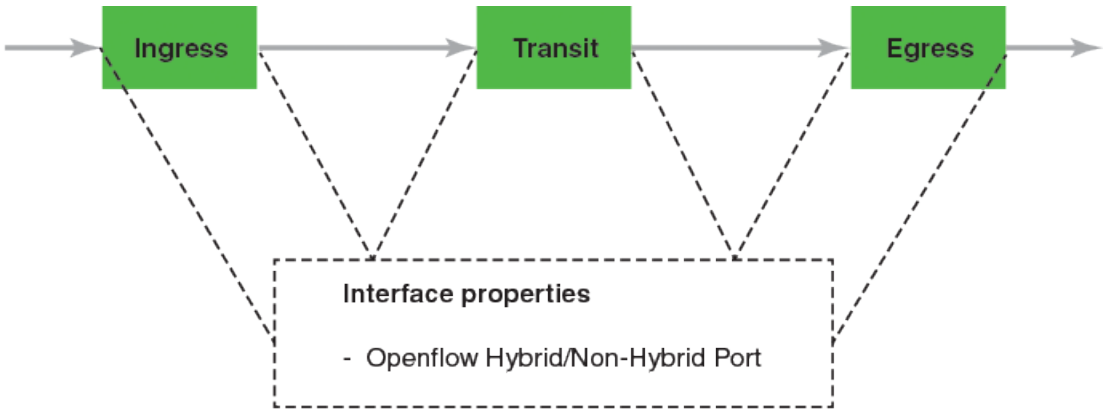
TABLE 32 Action taken at the node for L2VPN

Transit node	PHP node	Egress node
• Modify outer MPLS label • Flow – Match: MPLS label – Action: Swap MPLS label, Set EXP	• Pop outer MPLS label (Tunnel label) • Flow – Match: MPLS label – Action: Pop MPLS label, Set EXP	• Pop outer MPLS label (VC label) • Flow – Match: MPLS label – Action: Pop MPLS label

MPLS with OpenFlow infrastructure

Egress must be OpenFlow node for L3VPN and L2VPN. For IP over MPLS egress can be on a non-OpenFlow node.

FIGURE 13 Using OpenFlow flows from ingress to egress nodes



- IPv4-L3VPN: Ingress node pushes L3VPN VC label (OpenFlow MPLS label) and OpenFlow MPLS tunnel label using OpenFlow flow. Transit node matches on tunnel label and decides to swap or pop tunnel label. Egress node has OpenFlow flow matching on VC label pushed at ingress node.
- IP over MPLS: Ingress node pushes MPLS tunnel label using OpenFlow flow. Egress node packet is forwarded by native MPLS transit nodes. Transit node matches on tunnel label and decides to swap or pop tunnel label. Egress node has a flow matching on Layer 2 or Layer 3 fields.
- L2VPN: Ingress node pushes L2VPN VC label (OpenFlow MPLS label) and OpenFlow MPLS tunnel label using OpenFlow flow. Transit node matches on tunnel label and decides to swap or pop tunnel label. Egress node has OpenFlow flow matching on VC label pushed at ingress node.

MPLS label in flow match should be within local MPLS label range. MPLS label present in flow action can be any label (within configurable remote label range) which can swap or push. 128K MPLS labels (range is from 368928 to 499999) is reserved for OpenFlow from dynamic MPLS label range.

OpenFlow MPLS TTL propagation

Propagate TTL is based on existing global MPLS commands. TTL propagation from IP to MPLS is done by default for IP over MPLS. To disable TTL propagation, you can configure this command on ingress MPLS node:

```
device(config-mpls-policy)# no propagate-ttl
```

TTL propagation from IP to MPLS is not done for IPv4VPN by default. To enable TTL propagation, you can configure on ingress MPLS node using this command.

```
device(config-mpls-policy)# propagate-ttl
```

OpenFlow MPLS scaling

These are the supported numbers for MPLS flows for action and match fields on the nodes.

- Ingress:
  - 32K L2VPN flows - Flows that contain L2VPN label (remote label) in action.
  - 8K L3VPN or IPv4 over MPLS flows - Flows that contain non L2VPN label in action.
- Transit:
  - 8K transit MPLS flows.
- Egress:
  - 32K L2VPN flows - Flows that contain L2VPN label in match.

- 8K L3VPN or IPv4 over MPLS flows - Flows that contain non L2VPN label in match.

NOTE
You may be able to configure more number of MPLS flows than the above specified limit.

Statistics

The packet and byte count is supported for MPLS flows (MPLS label in action and match). Meter, enqueue and groups are also supported on MPLS flows.

Limitations and prerequisites

- A range of 128K label is reserved for OpenFlow, upper half for L2VPN and lower half for L3VPN/MPLS.
- MPLS supports maximum of two labels in the label stack.
- MPLS label is not supported for Ether type 0x8848.
- There is no differentiation for matching between Ether type 0x8847 and Ether type 0x8848 for incoming MPLS packets.
- VLAN is provided during pushing MPLS label.
- OpenFlow MPLS is supported on MLX Series with limitations (4x40GbE, 8x10GbE, 2x100GbE, 2x100GbE and 20x10GbE interface modules only).
- Maximum of 3K different next-hops (DA MACs) is used, if flows are pushing MPLS labels at ingress node for L2VPN, L3VPN or IPV4 over MPLS.
- Layer 3 IPv4 header is not modified except for TTL in ingress and egress nodes. Layer 3 header modification is not supported.
- Packets to controller may not have exact Layer 2 header, VLANs and MPLS header.
- VLAN preservation in not supported on MPLS flows (MPLS label in action and match).
- Port based flows matching MPLS label is not supported.

Configuring OpenFlow MPLS label range

To configure OpenFlow MPLS L2VPN remote label range, enter the following command:

```
device(config)#openflow ?
  controller         Configure controller
  default-behavior   Default forwarding for no match packets
  enable             Enable/disable OpenFlow
  mpls-l2vpn         Openflow Mpls l2vpn Config

devive(config)#openflow mpls-l2vpn  ?
remote-label-range   Configure the Remote label-ranges
```

To set the minimum value for the label range, enter the following command:

```
device(config)#openflow mpls-l2vpn remote-label-range ?
min-value   Min-value of the range
device(config)#openflow mpls-l2vpn remote-label-range min-value ?
DECIMAL   Range 1 - 1048575
```

To set the maximum value for the label range, enter the following command:

```
device(config)#openflow mpls-l2vpn remote-label-range min-value 500 ?
  max-value   Max-value of the range
device(config)#openflow mpls-l2vpn remote-label-range min-value 500 max-value ?
DECIMAL   Range 1 - 1048575
device(config)#openflow mpls-l2vpn remote-label-range min-value 500 max-value 1000 ?
```

MPLS label ranges on the OpenFlow

Here is an example to show the range of MPLS labels.

```
device#show openflow
Administrative Status:       Enabled
SSL Status:                  Enabled

Controller Type:             OFV 100
Number of Controllers:       1

Controller 1:
Connection Mode:       passive, TCP,
Listening Address:     0.0.0.0
Connection Port:       6633
Connection Status:      TCP_LISTENING


Match Capability:
L2/L3/L23: MPLS Label, MPLS BoS

Normal Openflow Enabled Ports:
Openflow Hybrid Interfaces:


Default action: drop
Maximum number of flows allowed: 4096
Active flow 0

Maximum number of Protected Vlans allowed: 1000
Maximum number of Unprotected Vlans allowed: 1000
Total number of Protected Vlans: 0
Total number of Active Protected Vlans: 0
Total number of Unprotected Vlans: 1
OpenFlow MPLS label range: 368928 - 499999
OpenFlow MPLS L3VPN or tunnel label range: 368928 - 434463
OpenFlow MPLS L2VPN label range: 434464 - 499999
OpenFlow MPLS L3VPN or tunnel remote label range: 368928 - 434463
OpenFlow MPLS L2VPN remote label range: 434464 - 499999
Total number of OpenFlow MPLS labels used: 8
Openflow IPSec logical port range internal(external): 1889(16385) - 2144(16640)
```

MPLS label match for the OpenFlow resources

Here is another example for MPLS label match flows shown.

```
device#show openflow resources
Openflow Resources:
CAM Profile: default
Used - Number of HW entries consumed
Free - Number of Port based flows that can be successfully programmed

Slot: 2         Module: BR-MLX-40Gx4-M 4-port 40GbE Module
Openflow Layer2or3 Flows   : MAX:   400          Used:     0     Free:   400
Openflow Layer23IPv4 Flows : MAX:   400          Used:     0     Free:   400
Openflow Layer23IPv6 Flows : MAX:   400          Used:     0     Free:   400

Slot: 3         Module: NI-XMR-10Gx4 4-port 10GbE Module
Openflow Layer2or3 Flows   : MAX:   400          Used:     0     Free:   400

Openflow MPLS L2VPN Label Match Flows           : MAX: 32768       Used:
0
Free: 65536
Openflow MPLS Tunnel/L3VPN Label Match Flows    : MAX: 8192      Used:     1
```

Supporting OpenFlow rules for the MPLS traffic

The MPLS traffic can follow the OpenFlow rules on an OpenFlow or hybrid OpenFlow Layer 2 or Layer23 ports without the Destination MAC address as the interface MAC address.

If this feature is disabled, irrespective of the Destination MAC address, the MPLS traffic on an OpenFlow Layer 2 or Layer23 port follows the OpenFlow MPLS rules.

Enabling OpenFlow MPLS traffic switched globally

This command allows you to make the incoming MPLS packets without the Destination MAC address as the interface MAC address to get switched and not hit the OpenFlow rules.

```
device(config)# openflow ?
  controller             Configure controller
  default-behavior       Default forwarding for no match packets
  enable                 Enable/disable OpenFlow
  hello-reply            Configure HELLO Reply for HELLO originated from
                         Controller
  mpls-l2vpn             OpenFlow MPLS L2VPN config
  mpls-l3vpn-or-tunnel   OpenFlow MPLS L3VPN or tunnel config
  mpls-us-enable         Native switching for packets without rcv interface mac
  <cr>
```

Group table

The group table introduces the ability to add support for port group abstraction for multi-pathing. This enables OpenFlow to represent a set of ports as a single entity for forwarding packets.

The group table supports the following group types:

- All: Executes all the buckets in the group; mostly used for flooding and multicasting.
- Indirect: Executes one defined bucket in the group. The action taken by this group type is sending packets to the next hop.
- Select: Executes one bucket in the group. The action bucket is chosen by a switch-defined algorithm, such as round robin or hashing (for example, load sharing).
- Fast failover: Executes the first live bucket, used in cases such as redundancy.

A group table consists of group entries. The counters in the following table are available in a group entry.

TABLE 33 Group entry counters

Counter	Description
Group Identifier	A 32-bit unsigned integer uniquely identifying the group
Group type	Determines group semantics
Counter	Number of packets processed by a group
Action bucket	Ordered list of action buckets, where each action bucket contains a set of actions to execute and associated parameters

Logical ports in port group is supported in this release. Logical ports are only supported on MLX Series.

Group messages

The following table describes the processing of group messages.

TABLE 34 Group messages

Group message type	Entry exists	Entry does not exist	Notes
Add (OFPGC_ADD)	Deny ADD. Return error message to controller	Add is processed	Subject to constraints below
Mod (OFPGC_MODIFY)	Group parameters and action buckets are updated	Deny MOD. Return error message to controller	Need to confirm if Update/ Modify is implemented as delete followed by add in the driver.
Del (OFPGC_DELETE)	Group entry is deleted. Flows which are associated with this group are also removed.	No Error. Message ignored	If a DEL comes in that has flows associated with it, then delete those flows from the system.

Error conditions and messages

This table lists the error conditions and the error OPCODES sent to the controller. The error type is always OFPET_GROUP_MOD_FAILED.

TABLE 35 Group messages

Error condition	Opcode
Adding group, if group already exists	OFPGMFC_GROUP_EXISTS
When group allocation exceeds memory or system limit	OFPGMFC_OUT_OF_GROUPS
Group type is not supported	OFPGMFC_BAD_TYPE
In case of group modification or deletion, if group does not exist	OFPGMFC_UNKNOWN_GROUP
Number of buckets in a group (each device has different limit)	OFPGMFC_OUT_OF_BUCKETS
Number of actions in a bucket greater than 1	OFPGMFC_BAD_BUCKET
Not an output port action	OFPGMFC_BAD_BUCKET

Multipart messages

Description	Opcode	Supported
Group counter statistics	OFPGMP_GROUP	Yes
Group description	OFPGMP_GROUP_DESC	Yes
List the capabilities of groups on a switch	OFPGMP_GROUP_FEATURES	Yes

Scaling group numbers

The output of the **show openflow groups** command displays the maximum number of actions in a bucket, the maximum number of buckets in a group and the maximum number of groups for scaling the group in OpenFlow.

```
device(config)# show openflow groups 20
```

On MLX and XMR Series devices

- The maximum number of actions in a bucket is 1.
- The maximum number of buckets in a group is 64.

- The maximum number of groups is 512.

For logical port group the maximum number of actions in a logical group bucket is 1, maximum number of buckets in a logical group is 8 and maximum number of logical groups is 128.

On CER 2000 Series and CES 2000 Series devices

- The maximum number of actions in a bucket is one.
- The maximum number of buckets in a group is 64; for group Select, it is 12.
- The maximum number of groups is 512; for group Select, it is 64.

Considerations and limitations for group tables

You must take into account the following when you configure group tables for OpenFlow flows.

For configuring group tables

- Devices support all group types in the OpenFlow v1.3.0 specification.
- The only action allowed in an action bucket is an output port; other supported actions can be included before a group table.
- Each action bucket can have only one output port.
- Groups cannot have physical port in one bucket and logical port in another, all buckets should either have physical ports or logical ports.
- Each OpenFlow logical port can be a part of any number of logical OpenFlow groups.
- A group entry can include ports from different slots and ports with different speeds.
- Group tables are not impacted based on the OpenFlow type on the interface (Layer 2 or Layer 3 or Layer23 and hybrid interfaces).
- To disable OpenFlow on interfaces, the interface must be removed from any group entry first.
- By default, there is a 22 Gbps ingress shaper per TM for multicast traffic. As port-group on device makes use of the multicast fabric queue, the data rate of the ingress shaper needs to be increased accordingly to ensure no packet drop at high data rate to port-groups.

For example, since each TM on 20x10GbE card has 100 Gbps forwarding performance, the following CLIs increase ingress shaper rate from the default of 22 Gbps to 100 Gbps to avoid packet drop on the card at the ingress.

```
device(config-if-e10000-12/1)# qos multicast shaper best-effort rate 100000000
device(config-if-e10000-12/1)# qos multicast shaper guaranteed rate 100000000
```

Limitations

For configuring OpenFlow, consider the following limitations.

- Maximum of 8 buckets are allowed in a OpenFlow group with logical ports.
- Group types All, Indirect and Fast failover are not supported for logical port groups. Logical port group only supports group type Select.
- Modification of group from all logical ports to all physical ports in buckets and vice versa are not supported.
- Watch_group is not supported in Fast failover group type.
- PBR or Transparent VLAN flooding cannot be configured along with the group table, when OpenFlow v1.3.0 is enabled and vice versa.

- If the ingress TM is configured to receive over 20 Gbps per traffic rate in a one to many port configuration, the Multicast Shaper command must be configured on the ingress port to the value of the actual ingress rate.

The following additional limitations apply to the specific group types.

For group All

To multicast flow-matching traffic to all action buckets, all action buckets are executed every time for the group All.

- A packet is replicated for the output port in each bucket. Only one packet is processed for each bucket of the group.

For group Indirect

The group Indirect executes one defined action bucket in a group. Only one action bucket can exist and it is executed every time.

Group Indirect supports one and only one bucket in each group entry.

For group Select

To load balance flow-matching traffic to all action buckets, one of the action buckets is chosen each time for the group Select.

- The only action allowed is output port.
- Weighted load balancing for group Select is not supported.
- Group chaining is not supported.
- Individual bucket statistics are not supported.

On CER 2000 Series and CES 2000 Series devices

The following additional limitations apply to the Select group type.

- A port can be part of only one group.
- All the ports in a group should be of same speed.
- Once a port has become the part of the group, then this physical port cannot be used in any flow as input port (if the port type is Layer 2).
- Once a port has become the part of the group, then this physical port cannot be used in any flow as output port.
- If a port is already part of some flow, then group addition and modification containing that port in a bucket is rejected.
- If an Layer 2 port is part of group Select, then generic flows is not accepted.

For group Fast failover

The group Fast failover executes the first live bucket. Each action bucket associated with a specific port or group determines the liveness of the bucket.

- The buckets are selected in the defined sequence.
- On a stack unit Fast failover, traffic convergence takes up to 3 seconds.
- If no buckets are live, packets are dropped.

Supporting flow formats for group

These are the supported flow formats on devices.

On MLX and XMR Series devices:

These devices only support one group in a flow. Flow can have multiple output ports. These are the formats supported for these devices.

o/p port: Output port

- [match] [modify actions] [group_id]
- [match] [modify actions] [o/p port_1] [group_id]
- [match] [modify actions] [o/p port_1] [o/p port_2].....[o/p port_n] [group_id]

Order of group_id and output ports does not make a difference. These are the formats not supported in this release.

- [match] [modify actions] [group_id_1] [group_id_2]
- [match] [modify actions] [group_id_1] [modify actions] [o/p port_1]

additional actions are not supported. Below flow formats will be supported. Multiple groups are not supported in one flow. Separate actions is not supported for each output port and group, if group exist in a flow.

Flow can have only logical port group in action. Along with logical group in action there can be either L2VPN or L3VPN label, any other additional actions are not supported. These are the flow formats supported.

- [match] [logical_group_id]
- [match] [logical_group_id] [L2VPN/L3VPN label]

On CER 2000 Series and CES 2000 Series devices:

These devices support multiple groups in one flow. This also supports separate actions for each output port and group in a flow.

- [match] [modify actions] [group_id]
- [match] [modify actions] [o/p port_1] [group_id]
- [match] [modify actions] [o/p port_1] [o/p port_2].....[o/p port_n] [group_id]
- [match] [modify actions] [group_id_1] [group_id_2]
- [match] [modify actions] [group_id_1] [modify actions] [o/p port_1]

Group events

These are the group events supported by OpenFlow.

The following group events are handled by OpenFlow control planes resided in Management Module to update group information to program hardware for forwarding information.

- Add group
- Delete group
- Add port to the group
- Delete port from the group
- Group type modification
- Group output port is up
- Group output port is down

Statistics

Group statistics are cumulative flow statistics that use the group ID in the action list. The following statistics are supported per group.

- Reference count (flow entries)
- Packet count (limited support on different devices)
- Byte count
- Duration (second)
- Duration (nanosecond)

For OpenFlow hybrid ports

- Group table does not affect hybrid functionality.
- Flows within a group on the hybrid port are treated the same as other flows.
- A group can support Normal and hybrid OpenFlow port together.

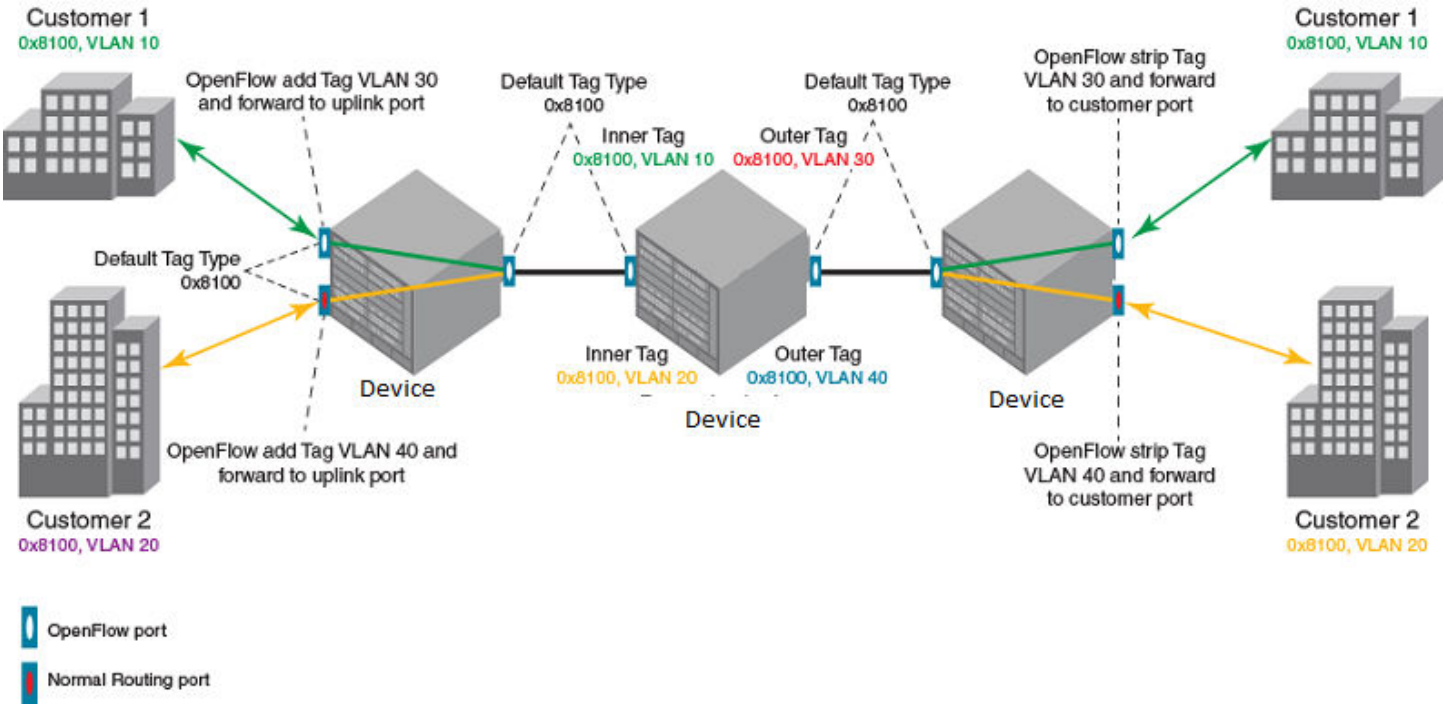
QinQ

You can push or pop or set VLAN tags in the outgoing packets of an OpenFlow flow with QinQ support. The ingress packet can be untagged, single tagged or double tagged frames. You can use QinQ to transport multiple customer segments or VLANs across Layer 2 infrastructures.

A OpenFlow flow matches on wild card VLAN or single VLAN and it does one of the following.

- No action
- Push outer VLAN
- Pop outer VLAN
- Modify or set outer VLAN

FIGURE 14 Transportation of VLAN or segment using QinQ



When a frame with QinQ tags arrives at the switch, the switching decision is made to match on the outer tag till the frame reaches a node configured to remove the outer tag.

On the customer ports, matching customer VLAN tag is supported. OpenFlow metering is also supported on flows with matching customer VLAN tags. Traditional features are supported for packets tagged with configured tag-type on the hybrid port.

- The customer VLANs are supported with hybrid functions on the hybrid ports with matching tag-type configuration.
- The customer packets can be forwarded by OpenFlow and traditional features.

NOTE
Matching fields in flow (VLAN + Layer 3 fields) are supported on Layer 3 and Layer23 OpenFlow Port.

NOTE
Matching fields in flow (VLAN + Layer 2 fields) are supported on Layer 2 and Layer23 OpenFlow Port.

On the uplink and transit port, OpenFlow can match outer VLAN tag. It supports addition, stripping or matching outer VLAN tag in Layer 2, Layer 3 and Layer23 mode. Hybrid port mode is supported for single tagged packets, but not for dual tagged packets. OpenFlow metering is also supported on flows with matching Outer VLAN tags. Traditional features are supported the packets tagged with configured tag-type on the hybrid port as well on uplink and transit ports.

Flow validation checks

Consider the following points when you configure OpenFlow v1.3.0 on devices for QinQ support.

- If the flow has more than one Push, Pop or Set VLAN field, the flow is not supported.
- If the flow includes two and above Push, Pop, or Set VLAN field, it is not supported.

- Layer 3 action fields cannot be supported in conjunction with Openflow actions Push and Pop VLAN, such flow is rejected. Similarly source and destination MAC modification action is not supported along with Push VLAN action.
- For every Push VLAN action there has to be a corresponding Set or Modify VLAN action field.

NOTE
Tag Type configuration is a port level setting, so all flows on a port recognize a packet as tagged only, if it is coming with the same Ether type as configured. Packets with other tag types are treated as untagged.

NOTE
Packets with more than two tags are dropped on a Layer 3 OpenFlow port.

These are the considerations for OpenFlow v1.3.0 QinQ support for CER 2000 Series and CES 2000 Series.

- Push, Pop or Modify one VLAN tag is supported in both OpenFlow Layer 2 and Layer23 (IPv4 and IPv6) mode.
- Push, Pop or Modify one VLAN tag can be combined with all existing OpenFlow actions including group, enqueue actions.
- One OpenFlow flow can have Push, Pop or Modify as one of the action in an action list.

QinQ action

These are the enhancements to the existing QinQ functionality.

- Push, Pop or Modify up to 2 VLAN tags per flow and up to 1 VLAN tag for CER 2000 Series and CES 2000 Series.
- Inner VLAN tag Ether type can be modified.

You are able to Push, Pop or Set up to 2 VLAN tags in the outgoing packets of an OpenFlow flow. The ingress packet can be untagged, single tagged or double tagged frames. You can use QinQ to transport multiple customer segments or VLANs across Layer 2 infrastructures.

TABLE 36 OpenFlow QinQ actions

OpenFlow action	Input traffic			
	Untagged	Single tagged	Double tagged	More than 2 tags
No Action	Egress traffic goes as untagged. (NO OP)	Single VLAN in ingress packet is preserved when sent out in egress direction. (NO OP)	Two VLANs in ingress packet are preserved when sent out in egress direction. (NO OP)	More than two VLAN tags in ingress packet are preserved when sent out in egress direction. (NO OP)
Push 2 VLANs	Go out as dual tagged packet at egress.	Go out with 3 tags at egress.	Go out with 4 tags at egress.	Go out with 5 tags at egress.
Pop 2 VLANS	Go out as untagged in egress direction. (NO OP)	Go out as untagged in egress direction as ingress VLAN tag is stripped.	Go out as untagged in egress direction.	Go out as single tagged(with outer and inner VLAN removed) packet at egress.
Modify/Set 2 VLANs	Go out as dual tagged packet at egress.	Go out as dual tagged packet at egress.	Go out as dual tagged packet at egress.	Go out as triple tagged packet at egress.

TABLE 37 OpenFlow QinQ actions for CER 2000 Series and CES 2000 Series

OpenFlow action	Input traffic		
	Untagged	Single tagged	More than one tag
No Action	Egress traffic goes as untagged. (NO OP)	VLAN tag in ingress packet is preserved. (NO OP)	All VLAN tags in ingress packet are preserved. (NO OP)

TABLE 37 OpenFlow QinQ actions for CER 2000 Series and CES 2000 Series (continued)

OpenFlow action	Input traffic		
	Untagged	Single tagged	More than one tag
Push Outer VLAN	Configured VLAN in the flow action is added in the egress direction.	Configured VLAN in the flow action is added as outer most VLAN in the egress direction in addition to existing VLAN.	Configured VLAN in the flow action is added as outer most VLAN in the egress direction in addition to existing VLAN.
Pop Outer VLAN	Go out as untagged in egress direction. (NO OP)	Go out as untagged in egress direction as ingress VLAN tag is stripped.	Go out as single tagged (with outer VLAN removed) packet at egress.
Modify/Set Outer VLAN	VLAN tag is added.	Outer VLAN tag is modified as per flow.	Outer VLAN tag is modified as per flow.

Inner VLAN Ether type can be modified to any user defined values. OpenFlow flow is rejected, if there is mismatch in tag-type of port and the push action.

Setting tag-type for Inner tag

The following tag-type command is modified to include an option to configure the tag-type for inner tags.

```
device(config)#tag-type ?
  HEX    etype in hex (Default: 0x8100)
  isid   Specify Etype for I-Tagged packets
  inner  Specify Etype for Inner Tag

device(config)#tag-type inner ?
  HEX    etype in hex (Default: 0x8100)

device(config)#tag-type inner <inner-tag-type> ?
  ethernet       Ethernet port
  pos            POS port

device(config)#tag-type inner <inner-tag-type> e
  ethernet   Ethernet port

device(config)#tag-type inner <inner-tag-type> ethernet ?
  SLOT/PORT   Interface number

device(config)#tag-type inner <inner-tag-type> ethernet <slot/port> ?
  ethernet       Ethernet port
  to             To an end range
  <cr>

device(config)#tag-type inner <inner-tag-type> ethernet <slot/port>
```

Show tag type

```
device(config)#show tag-type

VLAN TAG ETYPE
-------------------------------------------------------------
Global Tag-type 0x8100

        Port                Tag-Type        Inner-Tag-Type
        2/3                 0x88a8        0x8100


I-TAG ETYPE
-------------------------------------------------------------

Global I-TAG Tag-type 0x88e7

device(config)#tag-type tag1 ethernet 1/1
  ethernet       Ethernet port
  to            To an end range

device(config)#tag-type tag1 ethernet 1/1 to 1/2
  ethernet       Ethernet port
```

Configuring the tag value

```
device(config)#configure tag
  tag-type       Configure tag-types associated with tagged VLAN packets
  tag-value      Configure values for tag=type tag1, tag2 and isid

device(config)#tag-value
  isid       set isid tag-value
  tag1       set tag-type tag1 value
  tag2       set tag-type tag2 value
```

These are the examples for LP.

2 Push Action

New fields have been added for Outer and Inner VLAN ID.

```
device(config)#show openflow flows
-------------------------------------------------------------
Flow Id: 8, Priority: 32768, FD Id: 0, PW Id: 1
        Rule:
           In Port:      e2/1
           In Vlan:      Tagged[1210]
        Action: FORWARD
             Out Port:  e2/1, Push-vlan-tag 1: 0x8100, Vlan id: 3
   Push-vlan-tag 2: 0x8100, Vlan id: 4

           FID: -N/A-, MVID: -N/A-
        Hardware Information:
        Port: 2/1   PPCR Id : 3, CAM Index: 0x000576aa (L4)  PRAM Index: 0x0003ff63 Packets: 0
        Statistics:
           Total Pkts: 0
           Total Bytes: 0
```

2 Pop Action

```
device(config)#show openflow flows
-------------------------------------------------------------
Flow Id: 8, Priority: 32768, FD Id: 0, PW Id: 1
        Rule:
           In Port:      e2/1
           In Vlan:      Tagged[1210]
        Action: FORWARD
              Out Port:  e2/1, Pop-vlan-tag 1
    Pop-vlan-tag 2

           FID: -N/A-, MVID: -N/A-
        Hardware Information:
        Port: 2/1   PPCR Id : 3, CAM Index: 0x000576aa (L4)  PRAM Index: 0x0003ff63 Packets: 0
        Statistics:
           Total Pkts: 0
           Total Bytes: 0
```

2 Set Action

```
device(config)#show openflow flows
-------------------------------------------------------------
Flow Id: 8, Priority: 32768, FD Id: 0, PW Id: 1
        Rule:
           In Port:      e2/1
           In Vlan:      Tagged[1210]
        Action: FORWARD
              Out Port:  e2/1, Vlan id 1: 3
                         Vlan id 2: 3

           FID: -N/A-, MVID: -N/A-
        Hardware Information:
        Port: 2/1   PPCR Id : 3, CAM Index: 0x000576aa (L4)  PRAM Index: 0x0003ff63 Packets: 0
        Statistics:
           Total Pkts: 0
           Total Bytes: 0

Vlan id: 3
```

Push-vlan-tag and push-in-vlan-tag

```
device(config)#show openflow flows
-------------------------------------------------------------
Flow Id: 8, Priority: 32768, FD Id: 0, PW Id: 1
Rule:
In Port: e2/1
In Vlan: Tagged[1210]
Action: FORWARD
Out Port: e2/1, Push-vlan-tag 1: 0x8100, Vlan id: 3
Push-vlan-tag 2: 0x8100, Vlan id: 4
FID: -N/A-, MVID: -N/AHardware
Information:
Port: 2/1 PPCR Id : 3, CAM Index: 0x000576aa (L4) PRAM Index: 0x0003ff63
Packets: 0
Statistics:
Total Pkts: 0
Total Bytes: 0
```

These are the outputs for MP.

2 Push tag

```
device(config)#show openflow flows
------------------------------------------------------------
Total Number of data packets sent to controller:                    0
Total Number of data bytes sent to controller  :                    0

Total Number of Flows: 3
        Total Number of Port based Flows: 3
        Total Number of L2 Generic Flows: 0
        Total Number of L3 Generic Flows: 0
        Total Number of L2+L3 Generic Flows: 0
        Total Number of L23 Generic Flows: 0

Total Number of Hardware entries for flows: 3
        Total Number of Hardware entries for Port flow: 3
        Total Number of Hardware entries for Generic flow: 0

Total Number of Openflow interfaces: 4
        Total Number of L2  interfaces: 0
        Total Number of L3  interfaces: 2
        Total Number of L23 interfaces: 2

Flow ID: 5 Priority: 32768 Status: Active
        Rule:
           In Port:      e12/1
        Instructions: Apply-Actions
                Action: FORWARD
                        Out Port:  e12/2, Push-vlan-tag:         0x00008100 id: 100, Vlan PCP: 1
                                          Push-in-vlan-tag:      0x00008100 id: 200, In Vlan PCP: 2
        Statistics:
           Total Pkts: 0
           Total Bytes: 0
```

2 Pop tag

```
device(config)#show openflow flows
------------------------------------------------------------
Flow ID: 6 Priority: 32768 Status: Active
        Rule:
           In Port:      e12/2
        Instructions: Apply-Actions
                Action: FORWARD
                        Out Port:  e12/1, Pop-vlan-tag
                                          Pop-in-vlan-tag
        Statistics:
           Total Pkts: 0
           Total Bytes: 0
```

2 Set Action

```
device(config)#show openflow flows
------------------------------------------------------------
Flow ID: 7 Priority: 32768 Status: Active
        Rule:
           In Port:      e12/1
           In Vlan:      Tagged[100]
        Instructions: Apply-Actions
                Action: FORWARD
                        Out Port:  e12/2, Vlan: id: 500, Vlan PCP: 5
                                          In Vlan: id: 600, In Vlan PCP: 6
        Statistics:
           Total Pkts: 0
           Total Bytes: 0
```

Limitations

1. Push, Pop or Modify action cannot be combined in one single command.
2. For Multi-output port case, all output ports should have the same VLAN action.
3. Layer 3 field, Source MAC and Destination MAC modification cannot be supported along with Push VLAN action.
4. Push VLAN action is not supported on Layer 2 OpenFlow port on 10x24GbE interface modules.
5. Inner VLAN PCP modification is not supported when 2 VLANs are pushed or modified in the OpenFlow action.
6. VLAN preservation is not supported on 10x24GbE interface modules, thus modify or pop action is applicable only on outermost VLAN.

Enqueue

The controller is able to set up and configure queues and then map flows to a specific queue. The queue configuration sets the queue ID for a packet and determines the queue to be used for scheduling and forwarding the packet.

Queue configuration takes place outside the OpenFlow protocol based on weights for a particular queue using Weighted Round Robin (WRR) scheduling.

```
device(config-if-e10000-2/5)# show qos scheduler ethernet 2/5

Port    | Scheme   Type    Pri7  Pri6   Pri5  Pri4  Pri3  Pri2  Pri1  Pri0
--------+----------------+-----+-----+-----+-----+-----+-----+-----+-----
   2/5  | weighted Weight 10    10    20    10    10    20    10    10
```

There are two distinct parts that form the enqueue mechanism:

- Configuration
- Flow-queue mapping or forwarding

Assuming that a queue is already configured, you can associate a flow with an OFPAT_ENQUEUE action which forwards the packet through the specific queue on a port. Note that an enqueue action overrides any TOS or VLAN_PCP-related behavior that is potentially defined in the flow, but the packet is not changed or modified due to an enqueue. The devices support a total of 8 queues per port.

In case of stacking, queue 7 is reserved for stacking messages. Any queue set to 7 is reclassified to queue 6. When there is no stacking, the standalone queue set to 7 remains as 7.

Use case: OpenFlow meter and enqueue

QoS is usually implemented to provide appropriate levels of service to support Service Level Agreements (SLAs). You have the ability to meter and determine customer traffic according to the bandwidth guaranteed provided to the customer by way of a combination of OpenFlow v1.0.0 or v1.3.0 actions. The policing must be fine grained and flexible enough as supported by OpenFlow match semantics. For instance, the match criteria for rate limiting one application may be based on a VLAN tag and, for another application, it may be based on the Layer 4 UDP or TCP port. The confirm action sets the appropriate queue ID for the packets, while the exceed action may cause the traffic to be dropped in case of congestion or remarked to a lower priority and with a different queue ID. When the packet is forwarded to a port using the output action, the queue ID determines which queue attached to this port is used for scheduling and forwarding the packet.

Configuring OpenFlow enqueue

Queue configuration takes place outside the OpenFlow protocol, either through a command line tool or through an external dedicated configuration protocol.

The minimum guaranteed bandwidth is configured through assignment of weights for a particular queue (with WRR scheduling).

CER 2000 Series and CES 2000 Series require extended QoS mode configuration to support the OpenFlow enqueue feature on a Layer 2 OpenFlow port.

Complete the following steps to configure OpenFlow enqueue.

1. Enable queue statistics at the global level.

```
device (config) # statistics
device (config-statistics # tm-voq-collection
```

2. Configure WRR scheduling and weights for the queues at the egress.

```
device(config-if-e10000-2/5)# qos scheduler weighted 10 10 20 10 10 20 10 10
```

3. Configure Shaper configuration for the queues at the egress port (configuring maximum rate).

```
device(config-if-e10000-2/5)# qos shaper priority 3 3000
```

4. Disable the encode policy map at the egress port.

```
device(config-if-e10000-2/5)# qos pcp encode-policy off
device(config-if-e10000-2/5)# qos dscp encode-policy off
```

5. Configure priority queues from 8 to 4 or vice versa.

```
device(config)# system-max-tm-queues 4
```

The queues are now configured for forwarding actions. After the queues have been configured, flows can be mapped to queues and packets are forwarded through them.

Limitations

The following limitations apply to the enqueue:

- A flow can have a maximum of one queue ID which is applicable for all output port in the action list.
- OpenFlow flows with action as Set IP TOS or Set VLAN PCP cannot be supported simultaneously with enqueue configuration. Such a configuration is rejected.
- QoS functionality of hybrid traffic flowing through these ports is affected.
- OpenFlow queue statistics is actually retrieved from existing TM statistics. The limitation of TM statistics applies for the OpenFlow queue statistics as well.
- The OpenFlow queue statistics is not supported for multicast flows.

Queue statistics

The following are the statistics supported per queue.

TABLE 38 Queue statistics supported on devices

Queue statistics	MLX Series XMR Series	CER 2000 Series CES 2000 Series
Number of transmitted packets	Yes	No
Number of transmitted bytes	Yes	No
Transmit overrun error, number of packets dropped due to overrun	No	No
Duration (seconds), time queue has been alive	No	No
Duration (nanoseconds), time queue has been alive	No	No

NOTE

A flow can have a maximum of one queue ID and applicable for all output ports in the action list.

To display queue level statistics, use this command as following.

```
device(config)# show np qos statistics ethernet 2/1
Port 2/1
  Ingress counters:
    COS 0: packets 1918620              bytes 168838560
    COS 1: packets 0                    bytes 0
    COS 2: packets 0                    bytes 0
    COS 3: packets 0                    bytes 0
    COS 4: packets 0                    bytes 0
    COS 5: packets 0                    bytes 0
    COS 6: packets 0                    bytes 0
    COS 7: packets 0                    bytes 0
```

Displaying OpenFlow queues

The **show openflow queue** command displays the queues attached to egress ports.

Ensure that OpenFlow queueing is configured on the device. The user can associate a flow with an **OFPAT_ENQUEUE** action which forwards the packet through the specific queue on a port.

1. Return to global configuration mode.

2. Enter the **show openflow queues** command and specify the queue IDs.

```
device(config)# show openflow queues 2/5

Openflow Port    2/5
  Queue 0
   Min Rate: 107374200 bps         Max Rate: 858993600 bps
   Tx Packets: 0
   Tx Bytes: 0
Openflow Port    2/5
  Queue 1
   Min Rate: 107374200 bps         Max Rate: 858993600 bps
   Tx Packets: 0
   Tx Bytes: 0
Openflow Port    2/5
  Queue 2
   Min Rate: 214748400 bps         Max Rate: 858993600 bps
   Tx Packets: 0
   Tx Bytes: 0
Openflow Port    2/5
  Queue 3
   Min Rate: 107374200 bps         Max Rate: 858993600 bps
   Tx Packets: 0
   Tx Bytes: 0
Openflow Port    2/5
  Queue 4
   Min Rate: 107374200 bps         Max Rate: 858993600 bps
   Tx Packets: 0
   Tx Bytes: 0
Openflow Port    2/5
  Queue 5
   Min Rate: 214748400 bps         Max Rate: 858993600 bps
   Tx Packets: 0
   Tx Bytes: 0
Openflow Port    2/5
  Queue 6
   Min Rate: 107374200 bps         Max Rate: 858993600 bps
   Tx Packets: 0
   Tx Bytes: 0
Openflow Port    2/5
  Queue 7
   Min Rate: 107374200 bps         Max Rate: 858993600 bps
   Tx Packets: 0
   Tx Bytes: 0
```

The output displays the minimum and maximum traffic rates in each interface.

Metering

Per-flow metering measures and controls the rate of packets for each flow entry. Per-flow meters enable OpenFlow to implement simple QoS operations, such as rate-limiting, and can be combined with per-port queues to implement complex QoS frameworks, such as DiffServ.

Meters are attached directly to flow entries. Each meter can have one or more meter bands. Each meter band specifies the rate of the band applies and the way packets are processed (DROP or DIFFSERV). OpenFlow metering operation is similar to ingress rate limiting in a QoS operation.

NOTE

Metering is not supported on CER 2000 Series and CES 2000 Series.

A meter table consists of meter entries. The counters in the following table are available in the meter entry.

TABLE 39 Meter entry

Counter	Description
Meter Identifier	A 32-bit unsigned integer uniquely identifying the meter
Meter band	A list of meter bands, where each meter band specifies the rate of the band and the way to process the packet. Rate and burst size are based on the line rate of the data traffic in contrast to the information rate.
Counter	Number of packets processed by a meter

Packets are processed by a single meter band based on the current measured meter rate. The meter applies the meter band with the highest configured rate that is lower than the current measured rate. If the current rate is lower than any specified meter band rate, no meter band is applied.

TABLE 40 Meter bands supported on devices

Meter bands	Supported
DROP	Yes
DSCP_REMARK	Yes
EXPERIMENTER	No

Each band type contains the following meter configuration parameters from the controller:

- Rate value in kbps
- Rate value in packets per second
- Burst size
- Statistics collection

TABLE 41 Meter configuration parameters

Configuration flags	Supported
OFPMF_KBPS	Yes
OFPMF_PKTPS	No
OFPMF_BURST	Yes
OFPMF_STATS	Yes

The metering system supports the features in the following table.

TABLE 42 Metering capabilities supported for metering features

Feature	MLX Series XMR Series
Maximum meter available in the system	3959
Band types (bitmap)	DROP, DSCP_REMARK
Capabilities (bitmap)	KBPS, BURST, STATS
Maximum number of band per meter	1
Maximum color value	3 (RED, YELLOW, GREEN)

Meter statistics

The following statistics are supported per meter:

- Flow count (number of flows associated with the meter)
- Input byte count (cumulative byte count on all associated flows)
- Duration (second)
- Duration (nanosecond)

The flow and the byte count calculate all packets processed by the meter. The duration fields indicate the elapsed time for which the meter has been installed on the device.

The byte band count counter is supported for the meter band type.

The byte band count presents the total numbers for all bytes processed by the band.

TABLE 43 Meter band statistics

Band Type	Supported
DROP	Yes
DSCP_REMARK	No

Limitations

The following limitations apply to the devices for metering:

On MLX Series and XMR Series devices

- A Meter cannot be applied with DSCP remark meter band alone.
- A maximum of 3959 non-priority based meters can be created in this device provided other application do not use it, as it is a shared resource and a maximum of 989 priority-based meters can be supported. These are shared resources with other application.
- When flows associated to a meter are all from different device in a device, applying exact meter rate is not possible.
- Packet count is not supported for individual meter and meter bands.
- Meters are not supported on flows with action as DROP or CONTROLLER.

On CER 2000 Series and CES 2000 Series devices

- If a flow is associated with a meter, flow statistics is not supported. The meter statistics is not substituted for flow statistics as the meter can be associated with more than one flow.
- Meter and counter share same hardware resources. The maximum resources available are 2K. These can be used either as meters or as counters (flow statistics).
- These 2K resources are shared with all other applications.
- If you program flows without a meter, it will, by default, consume a counter or meter resource. It is recommended that you configure all meters before configuring any flows to the system.

Meter band

The following limitations apply to the meter bands:

- The minimum burst size for DSCP or DROP band is 10 Kilobits.
- The maximum rate for DROP or DSCP is 100,000,000 kbps; the minimum is 0 kbps.
- The DSCP band rate cannot be greater than DROP band rate.
- The precedence level for DSCP band type should be from 0 through 63.

Displaying OpenFlow meters

A meter measures the rate of packets assigned to it and enables controlling the rate of those packets.

Return to global configuration mode.

The hardware resources are shared between OpenFlow and other features, so these resources are allocated on a first-come-first-serve basis.

Enter the **show openflow meters** command to showing all the meters in a flow for MP.

```
device(config)# show openflow meters 2
Meter id: 2

  Transaction id:        1438
  Meter Flags:            KBPS BURST STATS
  Flow Count:            0
  Number of  bands:      2
  In packet count:       -NA-
  In byte count:         0

  Band Type:    DSCP-REMARK

    Rate:                          750000
    Burst size:                    1500          kb
    Prec level:                    1
    In packet band count:          -NA-
    In byte band count:            0


  Band Type:    DROP

    Rate:                          1000000
    Burst size:                    2000          kb
    In packet band count:          -NA-
    In byte band count:            0

----
Total no. of entries printed: 1
```

The following example output shows a specific meter for LP.

```
device(config)# show openflow meters 1
Meter id: 1023

  Meter Flags:                   KBPS BURST
  Number of  bands:             2
  RL Class Index:               33      33
  In packet count:              -NA-
  In byte count:                0

  Band Type:    DROP

    Rate:                         3000         Adjusted rate:2996
    Burst size:                   1250          kb
    In packet band count:         -NA-
    In byte band count:           0

Band Type:    DSCP-REMARK

    Rate:                         1700         Adjusted rate:1693
    Burst size:                   1250          kb
    Prec level:                   27
    In packet band count:         -NA-
    In byte band count:           0
```

Meter implementation does not address any vendor specific proprietary messages.

TTL support

TTL-based actions are supported in the OpenFlow flows.

The system supports decrement TTL-based actions in OpenFlow flow for MLX Series and XMR Seriesdevices and CER 2000 Seriesand CES 2000 Series devices. Decrement TTL action is supported if flow is having match with ether type 0x800. It only supports Set TTL based action for CER 2000 Seriesand CES 2000 Seriesdevices.

show openflow flow has a field for SET TTL or DEC TTL in the flow action display.

Forwarding port and controller for matching ARP

If the flow has ARP as Ether type match, action can have output port or group along with send to controller option.

The following example shows openFlow flows output on MP.

```
ovs-ofctl add-flow tcp:10.25.123.245 --protocols=OpenFlow13 "in_port=145 dl_type=0x0806 actions=output:
152,controller"


Flow ID: 6 Priority: 32768 Status: Active
        Rule:
           In Port:      e4/1
           Ether type:   0x00000806
        Instructions: Apply-Actions
                Action: FORWARD
                        Out Port:  e4/8

                        Out Port: send to controller
```

It is supported on all Layer 2, Layer 3 and Layer23 OpenFlow ports. Packets sent to the controller (CPU) is rate-limited to maximum 500 packets/sec. This number is shared between different flows having action as send to controller. Along with ARP, other Layer 2 fields can be supported in the match criteria.

NOTE
Only Layer 2 and Layer23 ports are supported on CER 2000 Series and CES 2000 Series.

Limitations

- Only Ether type as 0x0806 is supported for such type of flows.
- Flow Modify or Set action for this flow is rejected. Action in such flows can have only **output_port**, **out_group** or controller. It cannot change the incoming traffic for such flows.
- When ARP flow is pushed with action send to controller and port mirroring then the traffic is mirrored to port specified in flow and is not sent to controller.

When the flow below is pushed on the device:

```
dpctl tcp:172.20.8.81 flow-mod cmd=add,table=0,idle=0,hard=0,prio=100
in_port=1,eth_src=00:00:00:01:00:11,vlan_vid=501,vlan_pcp=7,eth_type=0x0806 apply:output=ctrl,output=3
```

The **show openflow flows** has the following output.

```
device#show openflow flows
Total Number of data packets sent to controller:                    0
Total Number of data bytes sent to controller  :                    0

Total Number of Flows: 1
        Total Number of Physical Port based Flows         : 1

        Total Number of L2 Generic Flows                  : 0
        Total Number of L3 Generic Flows                  : 0
        Total Number of L2+L3 Generic Flows               : 0
        Total Number of L23 Generic Flows                     : 0
        Total Number of MPLS L2VPN Generic Flows                : 0
        Total Number of MPLS Tunnel/L3VPN Generic Flows         : 0

Total Number of Hardware entries for flows: 1
        Total Number of Hardware entries for Port flow: 1
        Total Number of Hardware entries for Generic flow: 0

Total Number of Openflow interfaces: 12
        Total Number of L2 Physical interfaces: 1
        Total Number of L3 Physical interfaces: 0
        Total Number of L23 Physical interfaces: 3
        Total Number of Logical interfaces: 8

Flow ID: 2 Priority: 100 Status: Active
        Rule:
           In Port:      e1/1
           In Vlan:      Tagged[501]
           Vlan PCP:     7
           Source Mac:   0000.0001.0011
           Source Mac Mask:      ffff.ffff.ffff
           Ether type:   0x00000806
           Idle Timeout                          :     0 secs
           Hard Timeout                          :     0 secs
        Instructions: Apply-Actions
                Action: FORWARD
                        Out Port:  e1/3
                        Out Port: send to controller
        Statistics:
           Total Pkts: 3918012
           Total Bytes: 501505536
        Timing Info:
           Time Elapsed(Since Flow Added)       :     4 secs
           Time Elapsed(Since Last Packet Hit)  :     0 secs
```

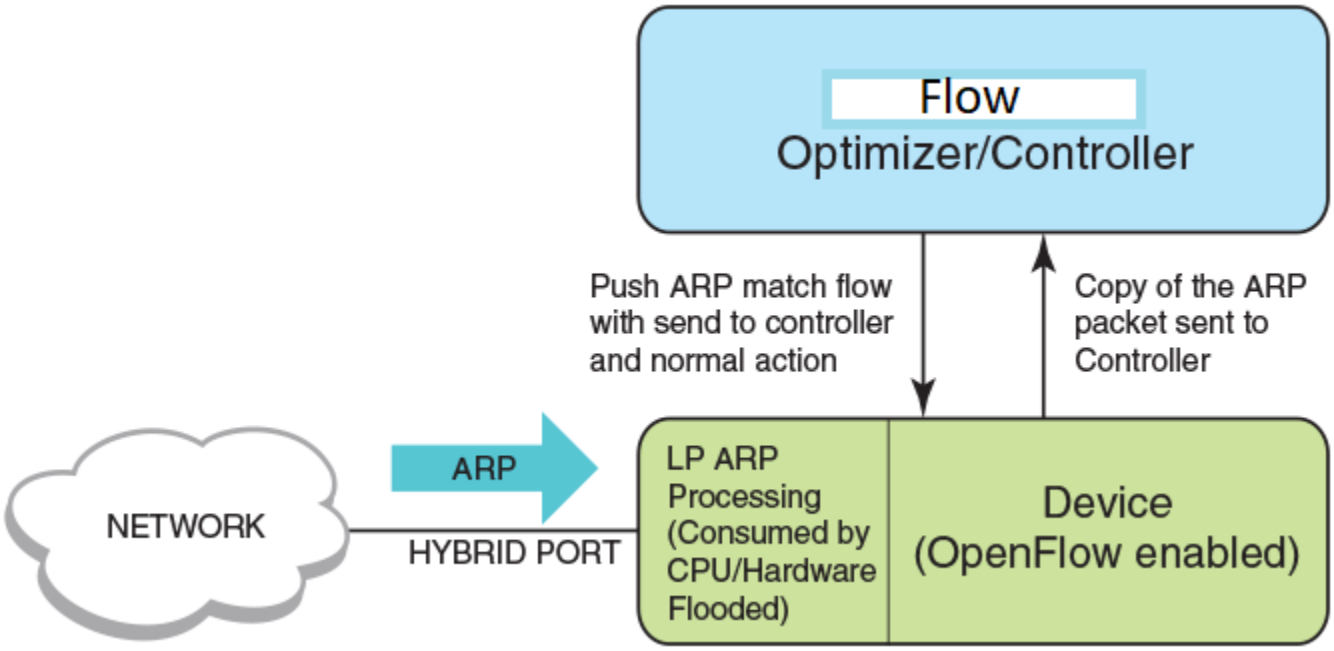
Supporting Normal action for flow matching on ARP packets

This feature supports the Normal action along with output as send to controller for flows matching on ARP packets.

OpenFlow rules are used to forward the ARP packets to the controller as well as to perform regular processing, such as consumed by CPU or hardware flooding in the VLAN domain.

ARP packets for both tagged and untagged traffic are supported on hybrid OpenFlow ports.

FIGURE 15 ARP packets processing



Configuration steps

Perform the following steps to configure this feature:

1. Configure OpenFlow globally.
2. Configure OpenFlow hybrid port.
3. Add a port-based flow matching VLAN and Ether type as 0x0806 along with action as Normal and send to controller.

The following condition applies for the flow installed with Normal and send to controller action.

- The flow with Normal and send to controller action together has higher priority than the flow with either send to controller or Normal action on the same port.

Configuration example

The following example configures ARP packet processing on the device.

```
ovs-ofctl add-flow tcp:10.25.122.210:6633
"in_port=147 dl_vlan=101 dl_type=0x806 actions=controller
 output:normal"--protocols=OpenFlow13

ovs-ofctl add-flow tcp:10.25.122.210:6633
"in_port=145 dl_type=0x806 actions=controller
 output:normal"--protocols=OpenFlow13
```

sFlow support on OpenFlow ports

sFlow is supported on OpenFlow enabled ports. sFlow samples both dropped and forwarded packets.

To support sFlow sampling on OpenFlow ports, the existing sFlow commands can be used on the OpenFlow port. All traffic on a protected VLAN, unprotected VLAN, and OpenFlow received on an OpenFlow port can be monitored together.

To enable sFlow forwarding on OpenFlow interfaces, you must do the following:

- Globally enable sFlow
- Enable sFlow forwarding on individual interfaces

```
device(config)# sflow enable
device(config)# interface ethernet 1/1 to 1/8
device(config-mif-1/1-1/8)# sflow forwarding
```

To change the sampling rate on an individual port, enter a command such as the following at the configuration level for the port.

```
device(config-if-e10000-1/1)# sflow sample 8192
```

To specify sFlow collectors, enter a command such as the following.

```
device(config)# sflow destination 10.10.10.1
device(config)# sflow destination ipv6 10:10::10:10
```

TLS1.2 support for NetIron devices

SSL or TLS is the Transport Layer Security protocol that provides a secure communications data channel for applications in NetIron devices such as HTTPS web server, syslog, TACACS+, and OpenFlow communications.

NetIron devices support the SSL V3.0, TLS1.0, and TLS1.1 protocols. To support the TLS1.2 protocol, the cryptographic engine has been updated.

The TLS1.2 protocol is functional in both server mode and client mode according to RFC 5246.

TLS1.2 in server mode

The HTTPS server communicates with the HTTPS clients or web browsers using the TLS1.2 protocol supporting all server functionality, such as user authentication.

The following popular web browsers are supported:

- Microsoft Internet Explorer
- Google Chrome
- Firefox

TLS1.2 in client mode

The NetIron device is a client and communicates with the remote server using the TLS1.2 protocol.

TLS1.2 in client mode supports the client certificate because the peer TLS1.2 server may request a certificate from the client.

The syslog and TACACS+ protection is enabled in the Common Criteria mode of operation.